Appendix

A Extended Related Work and Discussion

Parameter-Efficient Fine-tuning. Parameter-efficient tuning methods select a small subset of parameters or insert a few parameters to a pre-trained network. Then they only update the small subset of parameters, while keeping others fixed [7, 8, 9, 10, 11, 12, 13, 41]. For example, Adapters [13, 12] insert a small module into the transformer blocks and only update it. Similarly, prompt tuning [7] introduces a small vector that is concatenated with the input embeddings. BitFit [10] only tunes the bias term of the model. LoRA [11] injects trainable rank decomposition matrices into the transformer block. Although these methods are "parameter-efficient", they actually cannot reduce the memory usage of the model itself. This is because we still needs to build the computation graph for the whole model. Instead, the memory usage of optimizer states will be significantly reduced, which is in proportional to the number of trainable parameters [14].

Gradient Checkpointing. Gradient checkpointing helps decrease activation memory usage by saving only a selection of activations. However, it demands additional computation during the backward pass, as discarded activations must be recalculated [17, 16]. According to the report of Checkmate[2] [16], it achieves "a 2.3x memory reduction when training a BERT model with Checkmate optimizations (at 1x extra overhead for rematerialization)".

Limitations Although `WTA-CRS` significantly reduces the computation of the backward pass in a hardware-friendly way i.e., dropping entire rows/columns in the tensor, the current implementation still hampers the execution time of linear operations. This is because the extra sampling process and data movement counteract the acceleration. However, we note that (1) the overhead can be greatly reduced with better implementation, e.g., using prefetch and operation-fusion technique [28]; (2) the existing implementation can still yield a large speedup when employing larger batch sizes (Figure 9).

Potential Negative Societal Impacts. Our research primarily focuses on reducing the memory requirement of fine-tuning Language Models (LMs). The carbon emissions produced by LM fine-tuning may pose environmental issues. Our next step is to further improve the efficiency of LM fine-tuning, particularly on hardware with lower energy consumption.

B Unbiasedness of Weight Gradient

This part we directly follow the proof of Theorem 1 in ActNN [15]. For completeness, we provide the proof sketch here that is short and easy to follow. Specifically, here we use ReLU as the activation function for illustration convenience. We note that the conclusion in this section holds for any non-linear activation function. Specifically, the forward pass of ReLU-Linear at the l^{th} layer is

$$\begin{aligned}\mathbf{Z}^{(l+1)} &= \mathbf{H}^{(l)}\mathbf{W}^{(l)},\\ \mathbf{H}^{(l+1)} &= \mathrm{ReLU}(\mathbf{Z}^{(l+1)}),\end{aligned}$$

and the backward pass of ReLU is:

$$\begin{aligned}\mathbb{E}[\nabla\mathbf{Z}^{(l+1)}] &= \mathbb{E}[\mathbb{1}_{\mathbf{Z}^{(l+1)}>0} \odot \nabla\mathbf{H}^{(l+1)}]\\ &= \mathbb{1}_{\mathbf{Z}^{(l+1)}>0} \odot \mathbb{E}[\nabla\mathbf{H}^{(l+1)}],\end{aligned}$$

where $\odot$ is the element-wise product and $\mathbb{1}$ is the indicator function. The element-wise product is linear operation and $\mathbb{1}_{\mathbf{Z}^{(l+1)}>0}$ is only related to the pre-activation $\mathbf{Z}^{(l+1)}$ in the forward pass. We only apply the approximation during the backward pass so $\mathbb{1}_{\mathbf{Z}^{(l+1)}>0}$ can be extracted from the expectation. We know that for the last layer L, we have $\mathbb{E}[\nabla\mathbf{H}^{(L)}] = \mathbf{H}^{(L)}$ since we do not apply activation at the output layer. We then can prove by induction that $\mathbb{E}[\nabla\mathbf{H}^{(l+1)}] = \mathbf{H}^{(l+1)}$ and $\mathbb{E}[\nabla\mathbf{W}^{(l)}] = \mathbf{W}^{(l)}$ for any layer l.

[2] `https://github.com/parasj/checkmate/issues/153`

C Proof

C.1 Derivation of Equation (3)

Let $\mathbf{X} \in \mathbb{R}^{n\times m}$, $\mathbf{Y} \in \mathbb{R}^{m\times q}$ be two matrices. The matrix multiplication $\mathbf{XY}$ can be estimated as

$$\texttt{GEMM}(\mathbf{X},\mathbf{Y}) = \sum_{i=1}^{m} \mathbf{X}_{:,i}\mathbf{Y}_{i,:} \approx \sum_{t=1}^{k} \frac{1}{kp_{i_t}} \mathbf{X}_{:,i_t}\mathbf{Y}_{i_t,:} = \mathbf{X}'\mathbf{Y}',$$

Equation (3) shows the approximation error $\mathbb{E}[||\mathbf{XY} - \mathbf{X}'\mathbf{Y}'||_F]$ is minimized when the probabilities

$$p_i = \frac{||\mathbf{X}_{:,i}||_2 \; ||\mathbf{Y}_{i,:}||_2}{\sum_{j=1}^{m} ||\mathbf{X}_{:,j}||_2 \; ||\mathbf{Y}_{j,:}||_2}.$$

Proof. Let $f(i) = \frac{\mathbf{X}_{:i}\mathbf{Y}_{i:}}{p_i} \in \mathbb{R}^{n\times q}$. We note that $f(i)$ is an unbiased estimation of $\mathbf{XY}$. Namely,

$$\mathbb{E}_{j\sim\mathcal{P}}[f(j)] = \sum_{i=1}^{m} p_i \frac{\mathbf{X}_{:,i}\mathbf{Y}_{i:}}{p_i} = \mathbf{XY}.$$

Then we have

$$\mathbf{X}'\mathbf{Y}' = \frac{1}{k}\sum_{t=1}^{k} f(i_t), \tag{8}$$

where $i_1, \cdots, i_t$ are the index of the sampled column-row pairs at t^{th} random trials. For each i_t, its variance is

$$\begin{aligned}
\mathrm{Var}[f(i_t)] &= \mathrm{Var}[\frac{\mathbf{X}_{:i_t}\mathbf{Y}_{i_t:}}{p_{i_t}}] \\
&= \mathbb{E}[\frac{\mathbf{X}_{:i_t}^2\mathbf{Y}_{i_t:}^2}{p_{i_t}^2}] - \mathbb{E}^2[\frac{\mathbf{X}_{:i_t}\mathbf{Y}_{i_t:}}{p_{i_t}}] \\
&= \mathbb{E}[\frac{\mathbf{X}_{:i_t}^2\mathbf{Y}_{i_t:}^2}{p_{i_t}^2}] - (\mathbf{XY})^2. \\
&= \sum_{t=1}^{m} \frac{\mathbf{X}_{:t}^2\mathbf{Y}_{t:}^2}{p_t} - (\mathbf{XY})^2.
\end{aligned} \tag{9}$$

where the first step follows from the fact that $\mathrm{Var}[\mathbf{x}] = \mathbb{E}[\mathbf{x}^2] - \mathbb{E}^2[\mathbf{x}]$.

Then we have,

$$\begin{aligned}
\mathbb{E}[||\mathbf{XY} - \mathbf{X}'\mathbf{Y}'||_F] &= \sum_{i=1}^{n}\sum_{j=1}^{q} \mathbb{E}[(\mathbf{XY} - \mathbf{X}'\mathbf{Y}')_{ij}^2] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{q} \mathrm{Var}[(\mathbf{X}'\mathbf{Y}')_{ij}].
\end{aligned}$$

By combining Equation (8) and Equation (9) into the above equation, we have

$$\begin{aligned}
\mathbb{E}[||\mathbf{XY} - \mathbf{X}'\mathbf{Y}'||_F] &= \frac{1}{k}\sum_{i=1}^{n}\sum_{j=1}^{q}\sum_{t=1}^{m} \frac{\mathbf{X}_{it}^2\mathbf{Y}_{tj}^2}{p_t} - \frac{1}{k}\|\mathbf{XY}\|_F^2. \\
&= \frac{1}{k}\sum_{t=1}^{m} \frac{\|\mathbf{X}_{:,t}\|_2^2\|\mathbf{Y}_{t,:}\|_2^2}{p_t} - \frac{1}{k}\|\mathbf{XY}\|_F^2.
\end{aligned}$$

To minimize $\mathbb{E}[||\mathbf{XY} - \mathbf{X}'\mathbf{Y}'||_F]$, the optimal probability distribution can be obtained via solving the following optimization problem:

$$\min_{p_1,\cdots,p_m} \sum_{t=1}^{m} \frac{\|\mathbf{X}_{:,t}\|_2^2 \|\mathbf{Y}_{t,:}\|_2^2}{p_t},$$
$$\text{s.t.} \sum_{t=1}^{m} p_t = 1.$$

The solution to the above convex problem is the distribution defined in Equation (3). Namely,

$$p_i = \frac{||\mathbf{X}_{:,i}||_2 \ ||\mathbf{Y}_{i,:}||_2}{\sum_{j=1}^{m} ||\mathbf{X}_{:,j}||_2 \ ||\mathbf{Y}_{j,:}||_2}.$$

□

C.2 Unbiasedness of Our Proposed Estimator

Theorem 1 (Proof in Appendix C.2)**.** *The estimator defined in Equation (4) is an unbiased estimator for matrix production* $\mathbf{XY}$*, i.e,* $\mathbb{E}_{j\sim\mathcal{P}^{\mathcal{D}\setminus\mathcal{C}}}[\sum_{c\in\mathcal{C}} f(c)p_c + (1-\sum_{c\in\mathcal{C}} p_c)f(j)] = \mathbf{XY}$.

Proof.

$$\begin{aligned}
&\mathbb{E}_{j\sim\mathcal{P}^{\mathcal{D}\setminus\mathcal{C}}}\Big[\sum_{c\in\mathcal{C}} f(c)p_c + (1-\sum_{c\in\mathcal{C}} p_c)f(j)\Big] \\
=&\sum_{c\in\mathcal{C}} f(c)p_c + (1-\sum_{c\in\mathcal{C}} p_c)\mathbb{E}_{j\sim\mathcal{P}^{\mathcal{D}\setminus\mathcal{C}}}[f(j)] \\
=&\sum_{c\in\mathcal{C}} f(c)p_c + (1-\sum_{c\in\mathcal{C}} p_c)\sum_{j\in\mathcal{D}\setminus\mathcal{C}} \frac{p_j}{1-\sum_{c\in\mathcal{C}} p_c} f(j) \\
=&\sum_{c\in\mathcal{C}} f(c)p_c + \sum_{j\in\mathcal{D}\setminus\mathcal{C}} f(j)p_j \\
=&\mathbb{E}_{j\sim\mathcal{P}}[f(j)] \\
=&\mathbf{XY}
\end{aligned}$$

□

C.3 Variance of Our Proposed Estimator

Theorem 2 (Proof in Appendix C.3)**.** *Suppose the total budget of column-row pairs is k. If $\mathcal{C}$ satisfies*

$$\sum_{c\in\mathcal{C}} p_c > \frac{|\mathcal{C}|}{k}, \tag{7}$$

then we have $\mathrm{Var}[\hat{g}(\mathbf{X},\mathbf{Y})] < \mathrm{Var}[g(\mathbf{X},\mathbf{Y})]$*. Moreover,* $\mathrm{Var}[\hat{g}(\mathbf{X},\mathbf{Y})]$ *is minimized when* $|\mathcal{C}| = \min_{|\mathcal{C}|\in\{0,\cdots,k\}} \frac{1-\sum_{c\in\mathcal{C}} p_c}{k-|\mathcal{C}|}$.

Proof. Recall that the original estimator for matrix production $\mathbf{XY}$ is defined as

$$\mathbb{E}_{i\sim\mathcal{P}}[f(i)]. \tag{10}$$

and our proposed family of estimator is defined as:

$$h(j) = \mathbb{E}_{j\sim\mathcal{P}^{\mathcal{D}\setminus\mathcal{C}}}\Big[\sum_{c\in\mathcal{C}} f(c)p_c + (1-\sum_{c\in\mathcal{C}} p_c)f(j)\Big]. \tag{11}$$

We first define three independent random variables as belows:

$$u \sim \mathcal{P}^{\mathcal{C}}, \tag{12}$$

$$j \sim \mathcal{P}^{\mathcal{D}\backslash\mathcal{C}}, \tag{13}$$

$$b \sim \text{Bernoulli}(1 - \sum_{c \in \mathcal{C}} p_c). \tag{14}$$

According to the Law of total variance, we have

$$\begin{aligned} \text{Var}[f(i)] &= \mathbb{E}_b\Big[Var[f(i)|b]\Big] + \text{Var}_b\Big[\mathbb{E}[f(i)|b]\Big] \\ &\geq \mathbb{E}_b\Big[\text{Var}[f(i)|b]\Big] \\ &= \sum_{c \in \mathcal{C}} p_c \text{Var}[f(i)|b = 0] + (1 - \sum_{c \in \mathcal{C}} p_c)\text{Var}[f(i)|b = 1] \\ &\geq (1 - \sum_{c \in \mathcal{C}} p_c)\text{Var}[f(i)|i \in \mathcal{D}\backslash\mathcal{C}] \end{aligned} \tag{15}$$

where the first step follows from the fact that for any random variance $\mathbf{x}, \mathbf{y}$, we have $\text{Var}[\mathbf{y}] = \mathbb{E}[\text{Var}[\mathbf{y}|\mathbf{x}]] + \text{Var}[\mathbb{E}[\mathbf{y}|\mathbf{x}]]$. Also, by Equation (11), we have

$$\text{Var}[h(j)] = (1 - \sum_{c \in \mathcal{C}} p_c)^2 \text{Var}[f(j)|j \in \mathcal{D}\backslash\mathcal{C}]. \tag{16}$$

By combining the above two inequality, we have

$$\text{Var}[h(j)] \leq (1 - \sum_{c \in \mathcal{C}} p_c)\text{Var}[f(i)]. \tag{17}$$

Equation (17) quantitatively shows the variance reduction of $h(j)$ over $f(i)$. Then we compare our estimator $\hat{g}(\mathbf{X}, \mathbf{Y})$ and $g(\mathbf{X}, \mathbf{Y})$ in terms of variance.

First, because $g(\mathbf{X}, \mathbf{Y}) = \frac{1}{k}\sum_{t=1}^{k} f(i_t), \quad i_1, \cdots i_k \overset{\text{i.i.d}}{\sim} \mathcal{P}$. Thus we have

$$\text{Var}[g(\mathbf{X}, \mathbf{Y})] = \frac{1}{k}\text{Var}[f(i)]. \tag{18}$$

Similarly, we have

$$\text{Var}[\hat{g}(\mathbf{X}, \mathbf{Y})] = \frac{1}{k - |\mathcal{C}|}\text{Var}[h(j)]. \tag{19}$$

By combining Equation (17) into the above two equations, we have

$$\begin{aligned} \text{Var}[\hat{g}(\mathbf{X}, \mathbf{Y})] &= \frac{1}{k - |\mathcal{C}|}\text{Var}[h(j)] \\ &\leq \frac{1 - \sum_{c \in \mathcal{C}} p_c}{k - |\mathcal{C}|}\text{Var}[f(i)] \\ &\leq \frac{1 - \sum_{c \in \mathcal{C}} p_c}{k - |\mathcal{C}|} k\text{Var}[g(\mathbf{X}, \mathbf{Y})], \end{aligned} \tag{20}$$

where the first step follows from Equation (17). By setting $\frac{1 - \sum_{c \in \mathcal{C}} p_c}{k - |\mathcal{C}|} k \leq 1$, we arrive the conclusion that when $\sum_{c \in \mathcal{C}} p_c > \frac{|\mathcal{C}|}{k}$, we have $\text{Var}[\hat{g}(\mathbf{X}, \mathbf{Y})] \leq \text{Var}[g(\mathbf{X}, \mathbf{Y})]$.

Further, $\frac{1 - \sum_{c \in \mathcal{C}} p_c}{k - |\mathcal{C}|} k$ achieves the minimal when $|\mathcal{C}| = \min_{|\mathcal{C}| \in \{0, \cdots, k\}} \frac{1 - \sum_{c \in \mathcal{C}} p_c}{k - |\mathcal{C}|}$.

$\square$

D Implementation Details

The pseudocode for approximated linear layer with WTA-CRS and standard line layer is given in Algorithm 1 and Algorithm 3, respectively. The column-row pair sampling procedure is given in Algorithm 2. For the ease of illustration, we ignore the sequential length. As we mentioned in the main text, we only replace the GEMM in the backward pass with WTA-CRS . According to Equation (1c), we need the activation gradient $\nabla \mathbf{Z}$ to perform the column-row pair sampling during the forward pass. Thus we initialize a cache in CPU memory to store the gradient norm of activations from the last step. When performing column-row pair selection, we need to swap the gradient norm of activations between CPU and GPU, which will cause extra time overhead due to the data movement. Fortunately, we note that the number of elements in the gradient norm of activations is significantly less than the one in activations, which does not cause a significant time overhead.

Algorithm 1: Forward & Backward pass of Approximated Linear Layer

Hyperparameter: The total budget of column-row pairs k.
procedure INIT**:**
    Initialize Cache $\in \mathbb{R}^N$ as an empty matrix in main memory `//` N `is the total number of samples in the dataset. Cache is used for saving the norm of output gradient` $\nabla\mathbf{Z}$.
end
procedure FORWARD PASS**:**
    Input: activation $\mathbf{H} \in \mathbb{R}^{B\times D}$, weight $\mathbf{W} \in \mathbb{R}^{D\times D}$, indices of the current batch samples $BI = \{j_1, \cdots, j_B\}$.
    ctx $\leftarrow \{\}$ `// the context which saves tensors for backward`
    $\mathbf{Z} = \mathbf{HW}$
    $\mathbf{H}', ind \leftarrow$SUBSAMPLE($\mathbf{H}$, Cache$[BI]$, k)
    `// Cache`$[BI]$ `is the cached gradient norm from the backward pass;` ind `is the set of involved column-row pair indices`
    ctx $\leftarrow \{\mathbf{H}', \mathbf{W}, BI, ind\}$
    return $\mathbf{Z}$
end
procedure BACKWARD PASS**:**
    Input: ctx from the forward pass, output gradient $\nabla\mathbf{Z} \in \mathbb{R}^{B\times D}$
    $\mathbf{H}', \mathbf{W}, BI, ind \leftarrow$ ctx
    $\nabla\mathbf{H} = \nabla\mathbf{Z}\mathbf{W}^\top$
    $\nabla\mathbf{Z}' \leftarrow \nabla\mathbf{Z}[ind]$
    `//` $\nabla\mathbf{Z}' \in \mathbb{R}^{k\times D}$
    $\nabla\mathbf{W} = \mathbf{H}'^\top \nabla\mathbf{Z}'$
    for j in BI **do**
        Cache$[j]$ = $\|\nabla\mathbf{Z}_{j,:}\|_2$
    end
    `// Update the gradient norm of samples in the current batch`
    return $\nabla\mathbf{H}, \nabla\mathbf{W}$
end

Algorithm 2: SUBSAMPLE

Input: activation $\mathbf{H} \in \mathbb{R}^{B\times D}$, gradient norm $\mathbf{z} \in \mathbb{R}^{B}$, the total budget of column-row pairs k.

for $i = 1, \cdots, B$ **do**

  $p_i \leftarrow \frac{\mathbf{z}_i ||\mathbf{H}_{i,:}||_2}{\sum_{j=1}^{B} \mathbf{z}_i ||\mathbf{H}_{j,:}||_2}$ `// The probability of column-row pairs defined in Equation (3).`

end

$\hat{k} \leftarrow \min_{\hat{k}\in\{0,\cdots,k\}} \frac{1-\sum_{c\in\mathcal{C}} p_c}{k-\hat{k}}$, s.t. $\mathcal{C} = |\hat{k}|$. `//` $\mathcal{C}$ `is the set of column-row pair indices associated with` $|\mathcal{C}|$ `largest` p_i`.`

Sample $k - |\mathcal{C}|$ i.i.d. column-row pairs $\mathcal{C}_{\text{stoc}} = \{i_1, \cdots, i_{k-|\mathcal{C}|}\}$ from the distribution $\mathcal{P}^{\mathcal{D}\setminus\mathcal{C}}$

$ind \leftarrow \mathcal{C} \cup \mathcal{C}_{\text{stoc}}$

for $j \in \mathcal{C}_{stoc}$ **do**

  $\mathbf{H}[j,:] \leftarrow \mathbf{H}[j,:] * \frac{1-\sum_{c\in\mathcal{C}} p_c}{(k-|\mathcal{C}|)p_j}$ `// We need to normalize the stochastic part in Equation (6) to ensure the unbiasedness.`

end

$\mathbf{H}' \leftarrow \mathbf{H}[ind]$ `//` $\mathbf{H}' \in \mathbb{R}^{k\times D}$

return $\mathbf{H}'$, *ind*

Algorithm 3: Forward & Backward pass of the standard Linear layer

procedure FORWARD PASS**:**

  Input: activation $\mathbf{H}_Q \in \mathbb{R}^{BS\times D}$, weight $\mathbf{W}_Q \in \mathbb{R}^{D\times D}$, batch indices $index$

  `ctx` $\leftarrow \{\}$ `// the context which saves tensors for backward`

  $\mathbf{Z}_Q = \mathbf{H}_Q \mathbf{W}_Q$

  `ctx` $\leftarrow \{\mathbf{H}_Q, \mathbf{W}_Q\}$

  return $\mathbf{Z}_Q$

end

procedure BACKWARD PASS**:**

  Input: `ctx` from the forward pass, output gradient $\nabla\mathbf{Z}_Q$

  $\mathbf{H}_Q, \mathbf{W}_Q \leftarrow$ `ctx`

  $\nabla\mathbf{H}_Q = \nabla\mathbf{Z}_Q \mathbf{W}_Q^\top$

  $\nabla\mathbf{W}_Q = \mathbf{H}_Q^\top \nabla\mathbf{Z}_Q$

  return $\nabla\mathbf{H}_Q, \nabla\mathbf{W}_Q$

end

E More Experimental Results

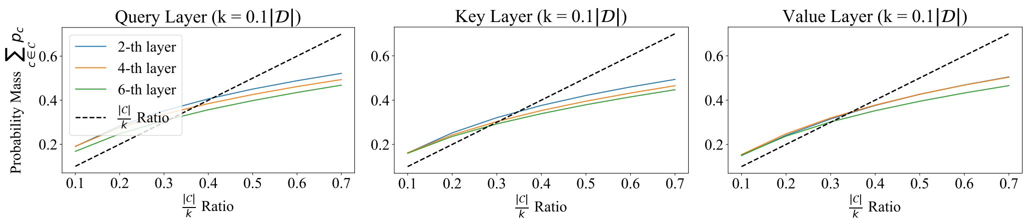


Fig. 10. The probability mass $\sum_{c\in\mathcal{C}} p_c$ versus $\frac{|\mathcal{C}|}{k}$ in Equation (7) at $k = 0.1|\mathcal{D}|$. Here we visualize the column-row index distribution of query/key/value layer T5-base model, fine-tuned on RTE dataset.

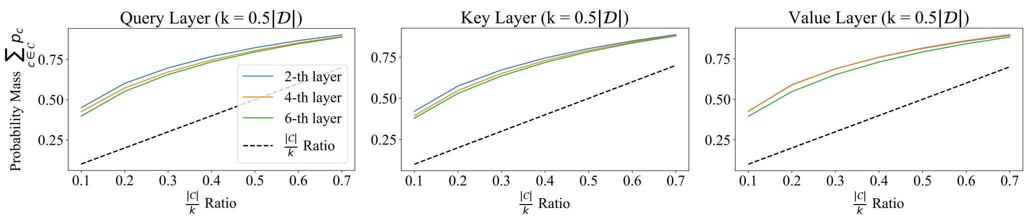


Fig. 11. The probability mass $\sum_{c\in\mathcal{C}} p_c$ versus $\frac{|\mathcal{C}|}{k}$ in Equation (7) at $k = 0.5|\mathcal{D}|$. Here we visualize the column-row index distribution of query/key/value layer T5-base model, fine-tuned on RTE dataset.

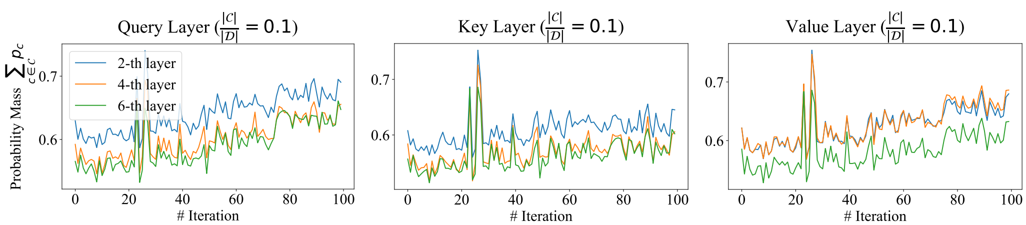


Fig. 12. The probability mass of top-10% column-row pairs in Equation (3) versus iterations. Here we visualize the query/key/value layer T5-base model, fine-tuned on RTE dataset.

E.1 More Experimental Analysis on Theorem 2

To evaluate Theorem 2 more comprehensively, below we also plot the $\sum_{c\in\mathcal{C}} p_c$ versus $\frac{|\mathcal{C}|}{k}$ at $k = 0.1|\mathcal{D}|$ and $k = 0.5|\mathcal{D}|$ in Figure 10 and 11, respectively. We also plot $\sum_{c\in\mathcal{C}} p_c$ versus iterations in Figure 12. We summarize that the the column-row index distribution is concentrated on a few column-row pairs. Thus, the assumption in Theorem 2 holds under the context of fine-tuning transformers.

E.2 More Experimental Speed Analysis

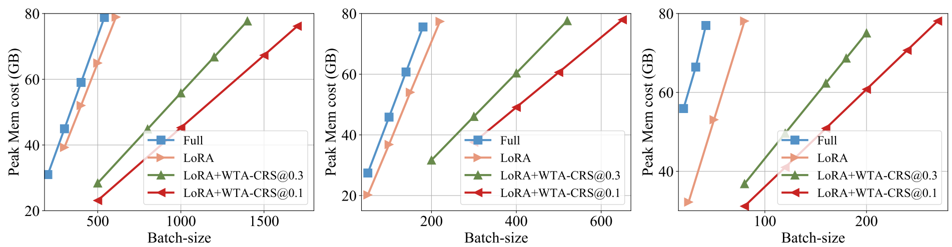
Increasing the batch size can often result in faster model convergence and/or enhance the final performance. Ideally, we should adjust the batch size according to the requirements of our model rather than being constrained by the GPU's memory capacity. To illustrate this, we have represented the correlation between peak memory usage and maximum mini-batch size for T5-Base, T5-Large, and T5-3B in Figure 13. Our observations highlight that `WTA-CRS` effectively increases the maximum available batch size.

We also provide the apple-to-apple speed comparison for linear operation with and without `WTA-CRS` in Table 3. In Table 3, "Fwd", "Bwd", and "F-B" are the time of forward pass, the time of backward pass, and the total time for both the forward and backward pass, respectively. We summarize that under the same workload, the current implementation of `WTA-CRS` may roughly slow down the linear

	Method	T5-ATT	T5-FF	T5 Block	T5-Large
Fwd	Full	8	10	17	1052
	`WTA-CRS`	22	16	37	2013
Bwd	Full	16	19	34	2073
	`WTA-CRS`	15	14	30	1738
F-B	Full	24	29	51	3125
	`WTA-CRS`	37	30	67	3751

Table 3: Latency (ms) of Forward and Backward pass.

operation about 20%. This is because the extra sampling process and data movement counteract the acceleration (see Algorithm 1). However, we note that (1) the overhead can be greatly reduced with better implementation, e.g., using prefetch and operation-fusion technique [28]; (2) the existing implementation can still yield a large speedup when employing larger batch sizes (Figure 9).



(a) T5-Base (b) T5-Large (c) T5-3B

Fig. 13. Peak memory usage versus maximum mini-batch size of T5.

F Experimental Settings

We give the detailed hyper-parameter setting in this section. Specifically, for both T5 and BERT models, the parameters are updated with the AdamW optimizer $\beta_1 = 0.9$ $\beta_2 = 0.999$ $\epsilon = 10^{-8}$ and weight decay $= 0$. The the learning rate is adjusted with a linear LR Scheduler, which maintains a constant learning rate for the initial 500 steps, and adjusts it gradually thereafter. The input sequences are padded to the maximum length 128. `WTA-CRS` has a LoRA dimension 32 if it is combined with LoRA. To achieve the optimal solution, the T5-Base, Large, 3B and BERT-Base and Large models have different learning rate, training epoch number, and mini-batch size on different datasets, which are given in Tables 5, 6, 7, respectively.

F.1 Computational Infrastructure

The computational infrastructure information is given in Table 4.

Table 4: Computing infrastructure for the experiments.

Device Attribute	Value
Computing infrastructure	GPU
GPU model	NVIDIA-A100
GPU Memory	81251MB
CUDA Version	11.4
CPU Memory	512GB

Table 5: Learning rate.

<table>
<tr><th>Model</th><th>Method</th><th>CoLA</th><th>SST-2</th><th>MRPC</th><th>QQP</th><th>MNLI</th><th>QNLI</th><th>RTE</th><th>STS-B</th></tr>
<tr><td rowspan="2">BERT-Base</td><td>WTA-CRS@0.3</td><td colspan="8">2e-5</td></tr>
<tr><td>LoRA+WTA-CRS@0.3</td><td>2e-4</td><td>5e-4</td><td>2e-4</td><td>3e-4</td><td>3e-4</td><td>2e-4</td><td>2e-4</td><td>3e-4</td></tr>
<tr><td rowspan="4">T5-Base</td><td>WTA-CRS@0.3</td><td colspan="5">3e-5</td><td>3e-6</td><td>3e-5</td><td>3e-5</td></tr>
<tr><td>WTA-CRS@0.1</td><td colspan="8">3e-5</td></tr>
<tr><td>LoRA+WTA-CRS@0.3</td><td>3e-4</td><td>3e-5</td><td>3e-4</td><td>3e-5</td><td>3e-5</td><td>3e-5</td><td>3e-4</td><td>3e-4</td></tr>
<tr><td>LoRA+WTA-CRS@0.1</td><td>3e-4</td><td>3e-5</td><td>3e-4</td><td>3e-5</td><td>3e-5</td><td>3e-5</td><td>3e-4</td><td>3e-4</td></tr>
<tr><td rowspan="2">BERT-Large</td><td>WTA-CRS@0.3</td><td colspan="8">2e-5</td></tr>
<tr><td>LoRA+WTA-CRS@0.3</td><td>3e-4</td><td>2e-4</td><td>2e-4</td><td>2e-4</td><td>2e-4</td><td>2e-4</td><td>3e-4</td><td>3e-4</td></tr>
<tr><td rowspan="4">T5-Large</td><td>WTA-CRS@0.3</td><td colspan="5">3e-5</td><td>3e-6</td><td>3e-5</td><td>3e-5</td></tr>
<tr><td>WTA-CRS@0.1</td><td colspan="5">3e-5</td><td>3e-6</td><td>3e-5</td><td>3e-5</td></tr>
<tr><td>LoRA+WTA-CRS@0.3</td><td>3e-4</td><td>3e-5</td><td>3e-4</td><td>3e-5</td><td>3e-5</td><td>3e-5</td><td>3e-4</td><td>3e-4</td></tr>
<tr><td>LoRA+WTA-CRS@0.1</td><td>3e-4</td><td>3e-5</td><td>3e-4</td><td>3e-5</td><td>3e-5</td><td>3e-5</td><td>3e-4</td><td>3e-4</td></tr>
<tr><td rowspan="2">T5-3B</td><td>LoRA+WTA-CRS@0.3</td><td>3e-4</td><td>3e-5</td><td>3e-4</td><td>3e-4</td><td>3e-4</td><td>3e-5</td><td>3e-4</td><td>3e-4</td></tr>
<tr><td>LoRA+WTA-CRS@0.1</td><td>3e-4</td><td>3e-5</td><td>3e-4</td><td>3e-4</td><td>3e-4</td><td>3e-5</td><td>3e-4</td><td>3e-4</td></tr>
</table>

Table 6: Training epoch number.

Model	Method	CoLA	SST-2	MRPC	QQP	MNLI	QNLI	RTE	STS-B
BERT-Base	WTA-CRS@0.3	20	20	10	10	10	10	20	10
	LoRA+WTA-CRS@0.3	60	20	20	20	20	20	40	40
T5-Base	WTA-CRS@0.3	40	10	20	10	10	10	50	20
	WTA-CRS@0.1	40	10	20	10	10	10	50	20
	LoRA+WTA-CRS@0.3	40	10	20	20	20	10	50	20
	LoRA+WTA-CRS@0.1	40	10	20	20	20	10	50	20
BERT-Large	WTA-CRS@0.3	60	20	20	10	10	10	40	10
	LoRA+WTA-CRS@0.3	60	20	20	20	20	20	40	40
T5-Large	WTA-CRS@0.3	20	10	20	10	10	10	40	20
	WTA-CRS@0.1	20	10	20	10	10	10	40	20
	LoRA+WTA-CRS@0.3	40	10	40	10	10	10	60	20
	LoRA+WTA-CRS@0.1	40	10	20	10	10	10	60	20
T5-3B	LoRA+WTA-CRS@0.3	40	10	20	10	10	10	60	20
	LoRA+WTA-CRS@0.1	40	10	20	10	10	10	60	20

Table 7: Training mini-batch size.

<table>
<tr><th>Model</th><th>Method</th><th>CoLA</th><th>SST-2</th><th>MRPC</th><th>QQP</th><th>MNLI</th><th>QNLI</th><th>RTE</th><th>STS-B</th></tr>
<tr><td rowspan="2">BERT-Base/Large</td><td>WTA-CRS@0.3</td><td colspan="7">128</td><td>16</td></tr>
<tr><td>LoRA+WTA-CRS@0.3</td><td colspan="7">128</td><td>16</td></tr>
<tr><td rowspan="4">T5-Base/Large/3B</td><td>WTA-CRS@0.3</td><td colspan="8">100</td></tr>
<tr><td>WTA-CRS@0.1</td><td colspan="8">100</td></tr>
<tr><td>LoRA+WTA-CRS@0.3</td><td colspan="8">100</td></tr>
<tr><td>LoRA+WTA-CRS@0.1</td><td colspan="8">100</td></tr>
</table>