Appendix for SparseBERT: Rethinking the Importance Analysis in Self-attention

A. Proof

A.1. Proof for Step 1

Lemma 2 (Lemma 8 (Yun et al., 2019)). *For any $f \in \mathcal{F}_{CD}$, there exists a piece-wise constant function $\bar{f}$ such that $d_p(f, \bar{f}) < \epsilon/3$.*

Proof. f is uniformly continously since f is a continous function on $[0,1]^{n\times d}$, which implies:
$\forall \epsilon > 0, \exists\, \delta > 0$, such that

$$\forall \boldsymbol{X}, \boldsymbol{Y}, ||\boldsymbol{X} - \boldsymbol{Y}||_\infty < \delta \Rightarrow ||f(\boldsymbol{X}) - f(\boldsymbol{Y})||_p < \epsilon/3.$$

Then we split the compact domain $[0,1]^{n\times d}$ into a grid of granularity δ, such that $G_\delta \in \{0, \delta, \ldots, 1\}$. By defining the following piece-wise constant function

$$\bar{f}(\boldsymbol{X}) = \sum_{\boldsymbol{G}\in G_\delta} f(\boldsymbol{G}) * \mathbb{1}\{\boldsymbol{X} \in \boldsymbol{G} + [0,\delta]^{n\times d}\},$$

we have

$$||f(\boldsymbol{X}) - \bar{f}(\boldsymbol{X})||_p = ||f(\boldsymbol{X}) - f(\boldsymbol{G})||_p < \epsilon/3.$$

Thus,

$$d_p(f, \bar{f}) = (\int ||f(\boldsymbol{X}) - \bar{f}(\boldsymbol{X})||_p^p d\boldsymbol{X})^{1/p} < \epsilon/3.$$

This proves the lemma. □

A.2. Proof for Step 2

A.2.1. Quantization (Feed-forward)

Lemma 3 (Lemma 5 (Yun et al., 2019)). *Consider a quantization mapping g_q^{ent}:*

$$g_q^{ent}(t) = \begin{cases} k\delta & \text{if } k\delta \le t < (k+1)\delta, k \in [1 : 1/\delta - 1], \\ -\delta^{-nd} & \text{otherwise.} \end{cases}$$

There exists a function g_q composed of $d/\delta + d$ token-wise feed-forward layers with $r = 1$ and piece-wise linear functions (at most three pieces), such that the quantization is performed on each entry of the input.

We first quantize the input $\boldsymbol{X}$ to their corresponding grid G_δ by quantization function g_q.

A.2.2. Contexual Mapping (Self-attention)

This is the main difference between ours and previous works, where self-attention without diag-attention adds additional constraints to the attention matrix. We will illustrate the definition of contextual mapping first and then prove Transformer blocks without diag-attention can also reach contextual mapping.

Definition 1. *(Contextual Mapping) For a set $G_\delta \in \mathbb{R}^{n\times d}$, a contextual mapping is a function mapping $q : G_\delta \to \mathbb{R}^n$ satisfying:*

- *For any $\boldsymbol{G} \in G_\delta$, all entries in $q(\boldsymbol{G})$ are distinct.*
- *For any $\boldsymbol{G}_1, \boldsymbol{G}_2 \in G_\delta$ ($\boldsymbol{G}_1 \neq \boldsymbol{G}_2$), all entries of $q(\boldsymbol{G}_1)$ and $q(\boldsymbol{G}_2)$ are distinct.*

Lemma 4. *There exists a function g_c composed of $\delta^{-d} + 1$ self-attention layers without diag-attention, such that $q(\boldsymbol{G}) := g_c(\boldsymbol{G})u$ satisfies the contextual mapping definition.*

Proof. Consider the function ψ, which can be implemented by a self-attention without diag-attention:

$$\begin{aligned}\psi(\boldsymbol{Z}; b)_i &= \sigma_H[(\boldsymbol{Z}_{i,:}\boldsymbol{u} - b)(\boldsymbol{Z}_{j\neq i,:}\boldsymbol{u})^\top]\boldsymbol{Z}_{j\neq i,:}\boldsymbol{u}\boldsymbol{e}^{(1)\top} \\ &= \begin{cases} (\max_{j\neq i} \boldsymbol{Z}_{j,:}\boldsymbol{u})\boldsymbol{e}^{(1)\top} & \text{if } \boldsymbol{Z}_{i,:}\boldsymbol{u} > b, \\ (\min_{j\neq i} \boldsymbol{Z}_{j,:}\boldsymbol{u})\boldsymbol{e}^{(1)\top} & \text{if } \boldsymbol{Z}_{i,:}\boldsymbol{u} < b, \end{cases}\end{aligned}$$

where $\boldsymbol{e}^{(1)} = [1, 0, \ldots, 0] \in \mathbb{R}^d$ and $\boldsymbol{u} \in \mathbb{R}^d$ is an auxiliary vector, which will be selected later.

We can contrust a self-attention layer without diag-attention that consists of two such heads $\Psi(\boldsymbol{Z}; b_1, b_2) = \psi(\boldsymbol{Z}; b_1) - \psi(\boldsymbol{Z}; b_2)$, such that

$$\Psi(\boldsymbol{Z}; b_1, b_2)_{i,1} = \begin{cases} \max_{j\neq i} \boldsymbol{Z}_{j,:}\boldsymbol{u} - \min_{j\neq i} \boldsymbol{Z}_{j,:}\boldsymbol{u} & \text{if } b_1 < \boldsymbol{Z}_{i,:}\boldsymbol{u} < b_2, \\ 0 & \text{otherwise.} \end{cases}$$

Thus, if we define a self-attention layer without diag-attention of the form $\boldsymbol{Z} \to \boldsymbol{Z} + \delta^{-d}\Psi(\boldsymbol{Z}; b_1, b_2)$, then selective shift operation is performed.

Next, we select $\boldsymbol{u} = (1, \delta^{-1}, \delta^{-2}, \ldots, \delta^{-d+1})$ and the following holds:

- If $Z_{i,j} \neq -\delta^{-nd}$ for all j, then $\boldsymbol{Z}_{i,:}\boldsymbol{u} \in [0 : \delta : \delta^{-d+1} - \delta]$. And the mapping from $\boldsymbol{Z} \in \{0, \delta, \dots, 1-\delta\}^d$ to $[0 : \delta : \delta^{-d+1} - \delta]$ is a bijective mapping.

- If $Z_{i,j} = -\delta^{-nd}$ for some j, then $\boldsymbol{Z}_{i,:}\boldsymbol{u} \leq -\delta^{-nd} + \delta^{-d+1} - \delta < 0$.

Thus, the mapping $\boldsymbol{Z}_{i,:} \to \boldsymbol{Z}_{i,:}\boldsymbol{u}$ is a bijective mapping for $\{0, \delta, \dots, 1-\delta\}^d$. We define $l_i = \boldsymbol{Z}_{i,:}\boldsymbol{u}$ and assume $l_1 < l_2 < \cdots < l_n$ without loss of generality.

For each $l \in [0 : \delta : \delta^{-d+1} - \delta]$, we choose $b_1 = l - \delta/2, b_2 = l + \delta/2$ and use δ^{-d} self-atteniton layers without diag-attention. Only one row will be in the range (b_1, b_2) each time and no other row will be affected. After above operation, l_i becomes $\widetilde{l_i}$ for better clarification.

For n rows, there are total n phases for column updating. After each i phases, we will maintain the following ordering:

$$l_{i+1} < l_{i+2} < \cdots < l_n < \widetilde{l_1} < \widetilde{l_2} < \cdots < \widetilde{l_i}.$$

Base Step When $i = 0$, it's the trivial case as

$$l_1 < l_2 < \cdots < l_n.$$

When $i = 1$, we have $\max_{j \neq 1} l_j = l_n$ and $\min_{j \neq 1} l_j = l_2$. $\widetilde{l_1} = \delta^{-d}(l_n - l_2) + l_1$.

$$\begin{aligned}
\widetilde{l_1} - l_n &= \delta^{-d}(l_n - l_2) + (l_1 - l_n) \\
&> \delta^{-d}(\delta) - (\delta^{-d+1} - \delta) \\
&= \delta^{-d+1} - \delta^{-d+1} + \delta \\
&= \delta > 0.
\end{aligned}$$

Inductive Step When $1 < i < n$, we have $\max_{j \neq i} l_j = \widetilde{l_{i-1}}$ and $\min_{j \neq i} l_j = l_{i+1}$. Thus, $\widetilde{l_i} = \delta^{-d}(\widetilde{l_{i-1}} - l_{i+1}) + l_i$. By expansion, we have:

$$\widetilde{l_i} = (l_n - l_2)\delta^{-id} + \sum_{j=1}^{i-1}(l_j - l_{j+2})\delta^{-(i-j)d} + l_i.$$

$$\begin{aligned}
\widetilde{l_i} - \widetilde{l}_{i-1} &= (l_n - l_2)(\delta^{-id} - \delta^{-(i-1)d}) \\
&+ \sum_{j=1}^{i-2}(l_j - l_{j+2})(\delta^{-(i-j)d} - \delta^{-(i-j-1)d}) \\
&+ \delta^{-d}(l_{i-1} - l_{i+1}) + l_i - l_{i-1} \\
&= (\delta^{-d} - 1)[(l_n - l_2)\delta^{-(i-1)d} + \sum_{j=1}^{i-2}(l_j - l_{j+2})\delta^{-(i-j-1)d}] \\
&+ \delta^{-d}(l_{i-1} - l_{i+1}) + l_i - l_{i-1} \\
&> (\delta^{-d} - 1)[\delta \cdot \delta^{-(i-1)d} + \sum_{j=1}^{i-2}(\delta - \delta^{-d+1})\delta^{-(i-j-1)d}] \\
&- \delta^{-d}(\delta^{-d+1} - \delta) + \delta \\
&= (\delta^{-d} - 1)\delta[\delta^{-(i-1)d} + \sum_{j=1}^{i-2}(1 - \delta^{-d})\delta^{-(i-j-1)d}] \\
&- \delta^{-d}(\delta^{-d+1} - \delta) + \delta \\
&= (\delta^{-d} - 1)\delta \cdot \delta^{-d} - \delta^{-d}(\delta^{-d+1} - \delta) + \delta \\
&= \delta > 0.
\end{aligned}$$

Therefore, $\widetilde{l_i} > \widetilde{l}_{i-1}$ holds and the ordering after operation on row i is:

$$l_{i+1} < l_{i+2} < \cdots < l_n < \widetilde{l_1} < \widetilde{l_2} < \cdots < \widetilde{l_i}.$$

When $i = n$, we have $\max_{j \neq i} l_j = \widetilde{l}_{n-1}$ and $\min_{j \neq i} l_j = \widetilde{l_1}$, resulting in $\widetilde{l_n} = \delta^{-d}(\widetilde{l}_{n-1} - \widetilde{l_1}) + l_n$. Similarly,

$$\begin{aligned}
\widetilde{l_n} - \widetilde{l}_{n-1} &= (l_n - l_2)(\delta^{-nd} - \delta^{-(n-1)d}) \\
&+ \sum_{j=1}^{n-2}(l_j - l_{j+2})(\delta^{-(n-j)d} - \delta^{-(n-j-1)d}) \\
&- \delta^{-2d}(l_n - l_2) + \delta^{-d}(l_{n-1} - l_1) + l_n - l_{n-1} \\
&= (l_n - l_2)(\delta^{-nd} - \delta^{-(n-1)d} - \delta^{-2d}) \\
&+ \sum_{j=1}^{n-2}(l_j - l_{j+2})(\delta^{-(n-j)d} - \delta^{-(n-j-1)d}) \\
&+ \delta^{-d}(l_{n-1} - l_1) + l_n - l_{n-1} \\
&> (\delta)(\delta^{-nd} - \delta^{-(n-1)d} - \delta^{-2d}) \\
&+ \sum_{j=1}^{n-2}(\delta - \delta^{-d+1})(\delta^{-(n-j)d} - \delta^{-(n-j-1)d}) \\
&+ \delta^{-d}(\delta) + \delta \\
&= \delta > 0.
\end{aligned}$$

The last inequation holds when $\delta^{-nd} - \delta^{-(n-1)d} - \delta^{-2d} > 0$, which is correct for $n > 2$ and small enough δ.

After n operations, we have $\widetilde{l}_1 < \widetilde{l}_2 < \cdots < \widetilde{l}_n$. Note that:

$$\begin{aligned}\widetilde{l}_n &= (l_n - l_2)\delta^{-nd} + \sum_{j=1}^{n-2}(l_j - l_{j+2})\delta^{-(n-j)d} \\ &\quad - (l_n - l_2)\delta^{-2d} + (l_{n-1} - l_1)\delta^{-d} + l_n \\ &< (\delta^{-d+1} - \delta)\delta^{-nd} + (\delta^{-d+1} - \delta)\delta^{-d} + (\delta^{-d+1} - \delta) \\ &= \delta(\delta^{-d} - 1)(\delta^{-nd} + \delta^{-d} + 1),\end{aligned}$$

thus $\widetilde{l}_i$ has the upper bound (denoted as Δ_h).

To ensure that all tokens are distinct, we will add two additional layers of the form $\boldsymbol{Z} \to \boldsymbol{Z} + (\Delta_h/\delta)\psi(\boldsymbol{Z}; 0)$.

First Global Shift Since $0 < \widetilde{l}_1 < \widetilde{l}_2 < \cdots < \widetilde{l}_n < \Delta_h$, the additional layer adds $(\Delta_h/\delta)(\max_{j\neq i} \boldsymbol{Z}_{j,:}\boldsymbol{u})\boldsymbol{e}^{(1)\top}$ for each i. Thus,

$$\widetilde{l}_i^+ = \begin{cases}\widetilde{l}_i + (\Delta_h/\delta)\widetilde{l}_n & \text{if } i \neq n, \\ \widetilde{l}_n + (\Delta_h/\delta)\widetilde{l}_{n-1} & \text{if } i = n.\end{cases}$$

For any $i, j \neq n$, we have $\widetilde{l}_i^+ < \widetilde{l}_j^+$ if $i < j$. Note that:

$$\begin{aligned}\widetilde{l}_1^+ - \widetilde{l}_n^+ &= (\Delta_h/\delta)(\widetilde{l}_n - \widetilde{l}_{n-1}) + \widetilde{l}_1 - \widetilde{l}_n \\ &> (\Delta_h/\delta)\cdot\delta - \Delta_h \\ &> 0,\end{aligned}$$

the order after first global shift is

$$\widetilde{l}_n^+ < \widetilde{l}_1^+ < \widetilde{l}_2^+ < \cdots < \widetilde{l}_{n-1}^+.$$

Second Global Shift At the second global shift, we have

$$\widetilde{l}_i^{++} = \begin{cases}\widetilde{l}_i^+ + (\Delta_h/\delta)\widetilde{l}_{n-1}^+ & \text{if } i \neq n-1, \\ \widetilde{l}_{n-1}^+ + (\Delta_h/\delta)\widetilde{l}_{n-2}^+ & \text{if } i = n-1.\end{cases}$$

By expansion, $\widetilde{l}_i^{++}$ has a more clear form as follows.

$$\widetilde{l}_i^{++} = \begin{cases}\widetilde{l}_{n-1} + (\Delta_h/\delta)(\widetilde{l}_{n-2} + \widetilde{l}_n) + (\Delta_h/\delta)^2\widetilde{l}_n & \text{if } i = n-1, \\ \widetilde{l}_n + 2(\Delta_h/\delta)\widetilde{l}_{n-1} + (\Delta_h/\delta)^2\widetilde{l}_n & \text{if } i = n, \\ \widetilde{l}_i + (\Delta_h/\delta)(\widetilde{l}_{n-1} + \widetilde{l}_n) + (\Delta_h/\delta)^2\widetilde{l}_n & \text{otherwise.}\end{cases}$$

The output of the second global shift is our g_c (i.e., $\widetilde{l}_i^{++} = g_c(\boldsymbol{G})_{i,:}\boldsymbol{u}$). Finally, we verify two properties of contextual mapping in Definition 1.

- For any $\boldsymbol{G} \in G_\delta$, we have $g_c(\boldsymbol{G})_{i,:}\boldsymbol{u} \mod (\Delta_h/\delta) = \widetilde{l}_i$. All entries of $q(\boldsymbol{G})\boldsymbol{u}$ are distinct because $\widetilde{l}_i$ is distinct with each other.
- For any $\boldsymbol{G}_1, \boldsymbol{G}_2 \in G_\delta$ ($\boldsymbol{G}_1 \neq \boldsymbol{G}_2$), each entry of $g_c(\boldsymbol{G}_i)\boldsymbol{u}$ lies in the interval $[(\Delta_h/\delta)^2\widetilde{l}_n, (\Delta_h/\delta)^2(\widetilde{l}_n + \delta))$. Since $\widetilde{l}_n$ is the unique identity for the input $\boldsymbol{G}$, all entries of $q(\boldsymbol{G}_1)$ and $q(\boldsymbol{G}_2)$ are distinct.

Therefore, $g_c(\boldsymbol{G})$ satisfies the definition of contextual mapping. □

A.2.3. Value Mapping (Feed-forward)

Lemma 5 (Lemma 7 (Yun et al., 2019))**.** *There exists a function g_v composed of $n(1/\delta)^{dn}$ token-wise feed-forward layers with $r = 1$ and piece-wise linear functions (at most three pieces), such that g_v is defined by a token-wise function g_v^{tkn},*

$$g_v(\boldsymbol{Z}) = [g_v^{tkn}(\boldsymbol{Z}_1) \dots g_v^{tkn}(\boldsymbol{Z}_n)],$$

where

$$g_v^{tkn}(\boldsymbol{Z}_i) = g_v^{tkn}(g_c(\boldsymbol{G})_i) = f(\boldsymbol{G}_i).$$

Therefore, we have $\bar{g}(\boldsymbol{X}) = g_v \circ g_c \circ g_q(\boldsymbol{X}) = \bar{f}(\boldsymbol{X})$ expect for a set has measure $O(\delta^d)$ (Yun et al., 2019), which implies that $d_p(\bar{f}, \bar{g}) \leq O(\delta^{d/p})$.

A.3. Proof for Step 3

Lemma 6 (Lemma 9 (Yun et al., 2019))**.** *For each modified Transformer blocks $\bar{g} \in \bar{\mathcal{T}}^{2,1,1}$, there exists the Transformer without diag-attention blocks $g \in \mathcal{T}^{2,1,4}$ such that $d_p(\bar{g}, g) \leq \epsilon/3$.*

Since the modification is only the softmax function and ReLU activation function (not related with the self-attention matrix A), the lemma still holds.

By Summarizing the above three steps, we have:

$$d_p(f, g) \leq d_p(f, \bar{f}) + d_p(\bar{f}, \bar{g}) + d_p(\bar{g}, g) \leq 2\epsilon/3 + O(\delta^{d/p}).$$

With enough small δ, we have $d_p(f, g) \leq \epsilon$. Thus, Transformers without diag-attention are also universal approximators.

B. Data Set

B.1. MNLI

The Multi-Genre Natural Language Inference (Williams et al., 2018) is a crowdsourced ternary classification task. Given a premise sentence and a hypothesis sentence, the target is to predict whether the last sentence is an [entailment], [contradiction], or [neutral] relationships with respect to the first one.

B.2. QQP

The Quora Question Pairs (Chen et al., 2018) is a binary classification task. Given two questions on Quora, the target is to determine whether these two asked questions are semantically equivalent or not.

B.3. QNLI

The Question Natural Language Inference (Wang et al., 2018b) is a binary classification task derived from the Stanford Question Answering Dataset (Rajpurkar et al., 2016). Given sentence pairs (question, sentence), the target is to predict whether the last sentence contains the correct answer to the question.

B.4. SST-2

The Stanford Sentiment Treebank (Socher et al., 2013) is a binary sentiment classification task for a single sentence. All sentences are extracted from movie reviews with human annotations of their sentiment.

B.5. CoLA

The Corpus of Linguistic Acceptability (Warstadt et al., 2019) is a binary classification task consisting of English acceptability judgments extracted from books and journal articles. Given a single sentence, the target is to determine whether the sentence is linguistically acceptable or not.

B.6. STS-B

The Semantic Textual Similarity Benchmark (Cer et al., 2017) is a regression task for predicting the similarity score (from 1 to 5) between a given sentence pair, whose sentence pairs are drawn from news headlines and other sources.

B.7. MRPC

The Microsoft Research Paraphrase Corpus (Dolan & Brockett, 2005) is a binary classification task. Given a sentence pair extracted from online news sources, the target is to determine whether the sentences in the pair are semantically equivalent.

B.8. RTE

The Recognizing Textual Entailment (Bentivogli et al., 2009) is a binary entailment classification task similar to MNLI, where [neutral] and [contradiction] relationships are classified into [not entailment].

B.9. SWAG

The Situations with Adversarial Generations (Zellers et al., 2018) is a multiple-choice task consisting of 113K questions about grounded situations. Given a source sentence, the task is to select the most possible one among four choices for sentence continuity.

B.10. SQuAD v1.1

The Stanford Question Answering Dataset (SQuAD v1.1) (Rajpurkar et al., 2016) is a large-scale question and answer task consisting of 100K question and answer pairs from more than 500 articles. Given a passage and the question from Wikipedia, the goal is to determine the start and the end token of the answer text.

B.11. SQuAD v2.0

The SQuAD v2.0 task (Rajpurkar et al., 2018) is the extension of above SQuAD v1.1, which contains the 100K questions in SQuAD v1.1 and 50K unanswerable questions. The existence of unanswerable question makes this task more realistic and challenging.

C. Implementation Details

The hyper-parameters of various downstream tasks are shown in Table 5.

Table 5. Hyper-parameters for different downstream tasks.

	GLUE	SWAG	SQuAD v1.1	SQuAD v2.0
Batch size	32	16	32	48
Weight decay	[0.1, 0.01]	[0.1, 0.01]	[0.1, 0.01]	[0.1, 0.01]
Warmup proportion	0.1	0.1	0.1	0.1
Learning rate decay	linear	linear	linear	linear
Training Epochs	3	3	3	2
Learning rate	[2e-5, 1e-5, 1.5e-5, 3e-5, 4e-5, 5e-5]			