Lossless 4-bit Quantization of Architecture Compressed Conformer ASR Systems on the 300-hr Switchboard Corpus

Zhaoqing Li\textsuperscript{1}, Tianzi Wang\textsuperscript{1}, Jiajun Deng\textsuperscript{1}, Junhao Xu\textsuperscript{1}, Shoukang Hu\textsuperscript{2}, and Xunying Liu\textsuperscript{1}

\textsuperscript{1}The Chinese University of Hong Kong; \textsuperscript{2}Nanyang Technological University
\textsupersetinline{\{zqli,twang,jjdeng,jhxu,xyliu\}@se.cuhk.edu.hk; \textsuperscript{2}shoukang.hu@ntu.edu.sg

Abstract

State-of-the-art end-to-end automatic speech recognition (ASR) systems are becoming increasingly complex and expensive for practical applications. This paper develops a high-performance and low-footprint 4-bit quantized Conformer ASR system. A key feature of the system design is to account for the fine-grained, varying performance sensitivity at different Conformer components to quantization errors. Neural architectural compression and mixed precision quantization approaches were used to auto-configure the optimal substructures and quantization bit-widths within each Conformer sub-module. Experiments conducted on the 300-hr Switchboard data suggest that the obtained auto-configured systems consistently outperform the uniform precision quantized baseline Conformer of comparable bit-widths in terms of word error rate (WER). An overall “lossless” compression ratio of 16.2 times was obtained over the 32-bit full-precision baseline while incurring no statistically significant WER increase.

\textbf{Index Terms:} speech recognition, model quantization, neural architecture search

1. Introduction

State-of-the-art end-to-end (E2E) automatic speech recognition (ASR) systems \cite{1, 2, 3, 4, 5, 6, 7} are becoming increasingly complex \cite{8, 9, 10} and expensive for practical applications. An ultimate goal for resource-intensive deep-learning-based AI applications, including E2E ASR systems, is to derive “lossless” model compression approaches that allow high-performance and low-footprint systems to be constructed\textsuperscript{1}. To this end, a powerful solution is to apply low-bit neural network quantization techniques, which have attracted intensive research interest in recent years and have successfully been applied to a range of AI applications, including computer vision \cite{12, 13, 14, 15, 16}, language modelling \cite{17, 18} and speech recognition systems \cite{19, 20, 21, 22}. Quantization methods replace deep neural network model parameters of full-precision floating points with quantized low-precision integer values. This allows model sizes and inference cost to be aggressively reduced for efficient implementations in FPGA systems.

Existing quantization studies for E2E ASR systems are predominantly uniform-precision-based. An identical setting of quantization bit width, for example, 4-bit, is applied to all the weight parameters \cite{12, 22, 23}. However, prior research on time delay neural network (TDNN) acoustic models \cite{24} and Transformer or LSTM-RNN language models \cite{18} have shown that the ASR system performance sensitivity to quantization errors varies among different internal components, and can be better accounted for using non-uniform, mixed precision approaches.

This paper aims to develop an ultra-compact 4-bit quantized Conformer-based E2E ASR system with high performance and low footprint on the 300-hr Switchboard corpus. In contrast to conventional uniform-precision-based quantization approaches, the overall system design of this paper accounts for the fine-grained, varying performance sensitivity of different model components to architecture compression and parameter quantization errors. The fundamental objective is to achieve the best model size versus performance trade-off operating points by locally optimizing each Conformer model component layer’s architecture size and quantization precision bit width. The overall system development comprises two major stages. In the first stage, the model structural redundancy is minimized using automated neural architectural compression techniques. Penalized differentiable architecture search \cite{25, 26} based neural architecture search (NAS) is applied to auto-configure the hidden layer dimensionality within each Conformer Encoder block. Inspired by the semi-orthogonal low-rank weight matrix factorization technique adopted in TDNN-F \cite{27}, the feed-forward sublayers’ weight parameter matrices are decomposed into two low-rank matrices. Their respective bottleneck dimensionalities are then auto-configured using NAS. The architectureurally compressed full-precision system serves as a streamlined starting point for the following mixed-precision quantization stage. The optimal local bit-width configurations are learned by minimizing the KL divergence measured sublayer-level performance sensitivity to quantization. Quantization-aware training (QAT) \cite{28, 29} is also used to reduce the performance degradation in the quantized parameter estimation stage after the local precision settings are determined.

The main contributions of the paper are summarized below:

1) To the best of our knowledge, this is the first work to apply neural architecture compression, low-rank weight factorization, and mixed precision quantization techniques to obtain ultra-compact Conformer ASR systems. Both these individual techniques and their combination have not been previously studied for Conformer ASR systems. Specifically, such combination was previously studied only with hybrid TDNN ASR systems \cite{24}. In terms of individual techniques, the KL divergence based automatically learned mixed precision quantization approach was only studied for hybrid TDNN systems \cite{24}, temporal convolutional network based speech separation \cite{30} and neural language models based on LSTMs and standard (non-Conformer) Transformers \cite{18}. Existing works on low bit precision quantization of Conformer (or other) ASR models largely focused on uniformly or manually configured quantization precision settings \cite{31, 32, 22, 33, 34, 35}. Other prior works on quantization for computer vision tasks \cite{12, 13, 14, 15, 16} lie

\textsuperscript{1}In all the experiments of this paper, “Lossless” compression is achieved when no statistically significant (MAPSSWE \cite{11}, $\alpha = 0.05$) WER increase is observed after model compression.
Figure 1: A paradigm of Conformer based end-to-end ASR system with neural architecture compression applied to the feed-forward modules (FFN) (in the green colored box) to locally select the dimensionality of either (a) the bottleneck subspace for their low-rank factorized weight matrices (left bottom); or (b) that of their hidden layer outputs (right bottom); or both. (c) auto-configuration of local bit-widths for each Encoder sub-module using the KL metric of Section 5.

2.1. Semi-orthogonal Low-rank Factorized Conformer
As the linear layers in the FFN module primarily account for the large model size of the Conformer. A simple way to compress the FFN modules is to reduce the hidden feature expansion dimensionality directly. Another option is inspired by the semi-orthogonal low-rank weight matrix factorization technique presented in [26, 27] for TDNNs. In [27], a large weight matrix is factorized to two smaller factors (i.e., projection and affine matrix), with one of the two constrained to be semi-orthogonal. For example, \(W = AB\), where \(W \in \mathbb{R}^{m \times n}\), \(A \in \mathbb{R}^{m \times k}\), and \(B \in \mathbb{R}^{k \times n}\), with \(B\) constrained to be semi-orthogonal, and \(k\) defines the bottleneck dimensionality. An illustration of Conformer FFN sublayer factorization is depicted in Fig. 1(a).

3. Neural Architecture Search
It is practically infeasible to manually search for an optimal configuration of the Conformer Encoder hidden layer or bottleneck dimensionality for large systems. In traditional studies, the commonly adopted uniform configuration fails to consider the varying performance sensitivity of different component configurations. To automatically learn a compressed and streamlined architecture configuration for the Conformer system, differentiable neural architecture search (DARTS) [25] is used to optimize the dimensionality of 1) their hidden layer outputs (Fig. 1(b)); or 2) that of the bottleneck projection subspace for their low-rank factorized weight matrices (Fig. 1(a)); or both 1) and 2). In DARTS, each architecture candidate in any layer is assigned a trainable architecture parameter \(\alpha\),

\[
h_l^i = \sum_{r=1}^{N_l} \lambda_l^i \phi_r = \sum_{r=1}^{N_l} \frac{\exp(\alpha_l^i)}{\sum_{r=1}^{N_l} \exp(\alpha_l^i)} \phi_r(h_l^{i-1}; \Theta_i),
\]

where \(\phi_r\) denotes the \(r\)-th candidate function of the \(l\)-th layer, \(\alpha_l^i\) and \(\Theta_i\) are corresponding architecture parameter and function weights, respectively. \(\lambda_l^i\) denotes architecture weight obtained by applying a Softmax function to the architecture parameter vector \(\alpha^i\). The output of the \(l\)-th layer \(h_l^i\) can be modeled as a weighted sum of all the candidates in the \(l\)-th layer. Then the architecture parameter and function weights can be trained in an alternative manner until convergence. Finally, the optimal architecture candidate at each layer is obtained according to \(\lambda\).

3.1. Gumbel-softmax DARTS
In traditional DARTS methods, the differences among architecture weights obtained by a flattened Softmax function are relatively small. This is prone to yield confusion over different candidates and thus leads to search errors. To minimize the confusion among different architectures, a Gumbel-softmax distribution [41, 42] is adopted to sharpen the architecture weights distribution to an approximately one-hot vector as,

\[
\lambda_l^i = \frac{\exp(\log(\alpha_l^i + g_l^i)/T)}{\sum_{j=1}^{N_l} \exp(\log(\alpha_l^i + g_j^i)/T)},
\]
Table 1: Performance of architecture compression of Conformer systems. “FW” and “FW-m” denote the hidden layer dimensionality of the first and the second feed forward layers in each EPN module, respectively. Similarly, “BN” and “BN-m” are corresponding bottleneck dimensionality for low-rank factorized FFNs. The hyper-parameter search scope for “FW” and “FW-m” hidden layer dimensions (Sys. 10-12) is from 1024 to 4096 with a step size of 128, denoted as indices [0, 24]. The hyper-parameter scope for “BN” and “BN-m” in Sys.13 is from 80 to 240 with a step size of 20 denoted as indices [0, 8], with “FW” and “FW-m” fixed as 2048. \( \{i, j\} \) denotes hyper-parameters from i-th layer to j-th layer inclusive set as “x” “.” “” is used as a delimiter between hidden and factorized bottleneck layers’ dimensionality settings. “S” and “L” denote low-rank weight matrices with and without semi-orthogonal constraints, respectively. \( \ast \) denote no significant (MAPSSWE [11], \( \alpha=0.05 \)) WER difference observed over the baseline system (sys.1).

<table>
<thead>
<tr>
<th>Sys.</th>
<th>Search Method</th>
<th>Search Object</th>
<th>( \eta )</th>
<th>Search Architecture</th>
<th>Hidden Dim</th>
<th>Beam Each</th>
<th>All</th>
<th>Encoder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Manual</td>
<td>-</td>
<td>-</td>
<td>FW (1 1 12) &amp; FW (m) (1 7 2) (40)</td>
<td>7.4</td>
<td>23791 11.7</td>
<td>602167 / 56BCM</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Manual</td>
<td>-</td>
<td>-</td>
<td>FW (1 1 12) &amp; FW (m) (1 7 2) (40)</td>
<td>7.6</td>
<td>15.5 11.8</td>
<td>31956 / 170BM</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Manual</td>
<td>-</td>
<td>-</td>
<td>FW (1 1 12) &amp; FW (m) (1 7 2) (40)</td>
<td>7.1</td>
<td>15.1 11.1</td>
<td>328141 / 99BCM</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Manual</td>
<td>-</td>
<td>-</td>
<td>FN (1 1 12) &amp; FW (m) (1 7 2) (40)</td>
<td>7.9</td>
<td>16.9 11.4</td>
<td>1202458 / 97.35BM</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Manual</td>
<td>-</td>
<td>-</td>
<td>FN (1 1 12) &amp; FW (m) (1 7 2) (40)</td>
<td>7.5</td>
<td>14.9 11.1</td>
<td>607947 / 47.88BM</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Manual</td>
<td>-</td>
<td>-</td>
<td>BN &amp; BN-m (1 1 12) (40)</td>
<td>7.4</td>
<td>15.7 11.1</td>
<td>607947 / 47.88BM</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Manual</td>
<td>-</td>
<td>-</td>
<td>BN &amp; BN-m (1 1 12) (40)</td>
<td>7.4</td>
<td>15.7 11.1</td>
<td>607947 / 47.88BM</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Manual</td>
<td>-</td>
<td>-</td>
<td>FN (1 1 12) &amp; FW (m) (1 7 2) (40)</td>
<td>7.2</td>
<td>15.1 11.1</td>
<td>607947 / 47.88BM</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Manual</td>
<td>-</td>
<td>-</td>
<td>FN (1 1 12) &amp; FW (m) (1 7 2) (40)</td>
<td>7.2</td>
<td>15.1 11.1</td>
<td>607947 / 47.88BM</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>PipeGumbel</td>
<td>FW</td>
<td>0</td>
<td>10.14 14 19 22 10 15 11 13</td>
<td>7.2</td>
<td>14.9 11.1</td>
<td>55104M / 42.10BM</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>PipeGumbel</td>
<td>FW-m</td>
<td>0.05</td>
<td>12.17 15 10 22 24 20 21 17 18 16 17</td>
<td>7.4</td>
<td>15.1 11.1</td>
<td>34543M / 21.63BM</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>PipeSoftmax</td>
<td>FW</td>
<td>0.01</td>
<td>0.012 12 0.122</td>
<td>7.4</td>
<td>15.7 11.1</td>
<td>34813M / 21.90BM</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>PipeSoftmax</td>
<td>BN</td>
<td>0.03</td>
<td>0.005 0.004 0.001</td>
<td>7.4</td>
<td>15.7 11.1</td>
<td>34813M / 21.90BM</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>PipeGumbel</td>
<td>FW &amp; BN</td>
<td>0.03</td>
<td>0.66 0.62 0.84 0.62 0.24 0.20 0.2</td>
<td>7.3</td>
<td>15.7 11.1</td>
<td>35015M / 22.10BM</td>
<td></td>
</tr>
</tbody>
</table>

where \( G_i = \log(- \log(U_i)) \) is the Gumbel variables and \( U_i \) is a uniform random variable. Eq. (3) approaches a categorical distribution as the temperature \( T \) decreases to zero.

3.2. Pipelined and Penalized DARTS

To avoid Gumbel-Softmax DARTS systems prematurely selecting sub-optimal architectures at an early stage, a pipeline style DARTS [43] is used to decouple the architecture parameter training from the model weights training process. Specifically, in pipelined DARTS, a supernet containing model parameters of all the candidates are trained to convergence at the first stage. The architecture parameters are then updated in the second stage on separate held-out data to avoid overfitting, with the other candidate internal parameters kept fixed.

During architecture compression, a penalty loss incorporating the complexity of each candidate choice is jointly optimized with the original Conformer loss function:

\[
\mathcal{L} = \mathcal{L}_{\text{conformer}} + \eta \sum_{i \leq n} \alpha_i C_i^f, \tag{4}
\]

where \( C_i^f \) is the complexity penalty term expressed as the number of parameters of the \( i \)-th candidate in the \( l \)-th layer. \( \eta \) is a trade-off hyper-parameter to be tuned.

4. Neural Network Quantization

Neural network quantization can further compress a neural network model by replacing the full precision weight parameters stored in 32-bit floating-point numbers with low-precision integers. An \( n \)-bit symmetric linear quantization can be denoted as:

\[
\hat{\Theta} = \Pi_{Q(\beta, n)}(\Theta), \tag{5}
\]

where \( \Theta \) and \( \hat{\Theta} \) are full-precision and quantized model weights, respectively. \( \Pi(\cdot) \) is a projection function that projects each element of \( \Theta \) to its closest value in a quantization table \( Q(\beta, n) \):

\[
Q(\beta, n) = \beta \times \{0, \pm 1, \pm 2, \ldots, \pm 2^{\beta-1} - 1\}, \tag{6}
\]

where \( \beta \) is a scaling factor to control the dynamic range of the weights and is stored separately as a full-precision floating-point number for the whole \( \Theta \). The most common choice is to set \( \beta = \max(|\Theta|) \), which is also adopted in this paper.

5. Mixed Precision Quantization

Different from uniform-precision quantization that fails to account for varying performance sensitivity of different layers (or components), the main idea of mixed-precision quantization is to keep more sensitive parts at higher precision while squeezing the bit widths more from less sensitive parts without increasing overall model size. The following Kullback–Leibler (KL) divergence metric serves to measure the output distribution difference between the original and quantized models [16, 24],

\[
\Omega_{t}(n) = \frac{1}{T} \sum_{t} D_{KL}(f(x_t, \hat{\Theta}) || f(x_t, \hat{\Theta}_{(n)})), \tag{7}
\]

where \( \Omega_{t}(n) \) denotes the sensitivity of the \( i \)-th layer to \( n \)-bit quantization. \( \hat{\Theta}_{(n)} \) denotes model parameters in the \( i \)-th layer is quantized to \( n \)-bit. \( f(\cdot) = \text{Softmax}(\phi(\cdot)) \) is to normalize the Encoder output into a distribution, \( x_t \) is the input vector at time step \( t \), among a total of \( T \) frames. The mixed precision settings for a \( L \)-layer Conformer system with a target model size \( S_{\text{target}} \) can be obtained by solving the following:

\[
\arg \min_{\{n_i\}_{i=1}^L} \sum_{i=1}^L \Omega_{i}(n_i) \text{ s.t. } \sum_{i=1}^L N_i + n_i \leq S_{\text{target}}, \tag{8}
\]

where \( N_i \) and \( n_i \) are the numbers of parameters and quantization bit width of the \( i \)-th layer. The optimization problem can be solved by a dynamic programming method to first obtain a Pareto frontier and then select the configuration with the lowest sensitivity that satisfies the model size constraint [16].

6. Experiments

The experiments are conducted on the 300-hr benchmark Switchboard corpus [44] with a single NVIDIA A40 GPU. The developed systems are evaluated on NIST Hub5’00, RT02, and RT03 evaluation sets containing 80, 120, and 144 speakers, and 3.8, 6.4, and 6.2 hours of speech, respectively. The baseline Conformer system is configured using the ESPNet [45] recipe.

As the Encoder module accounts for 81.5% of the baseline Conformer model parameters, architectural compression is performed on the Encoder only. In the quantization stage, both the Encoder and Decoder are low-bit quantized.

Experiments on neural architecture compression using complexity-aware NAS are shown in Tab. 1, where several trends can be found: 1) Before compressing, a series of manually re-configured Conformer systems (Sys. 1-9) are explored.

2ESPNet: egs/swbd/asr1/run.sh

3334
Table 2: Performance of the baseline full-precision, uniform-precision, and mixed-precision quantized Conformer systems with local precision settings automatically learned using the KL metric of Section 5 on NIST Hub5’00, RT02, and RT03. † denotes no significant (MAPSSWE [11], \(\alpha = 0.05\)) WER increase over the 32-bit baseline (ID 0). Speech processing real time factors (RTFs) in the last column.

<table>
<thead>
<tr>
<th>Sys.</th>
<th>ID</th>
<th>Quant. Pres.</th>
<th>Encoding Method</th>
<th>Bit Width</th>
<th>Conv. feed forward layer dimensionality</th>
<th>Hub5’00</th>
<th>RT02</th>
<th>RT03</th>
<th>Min. Enc. Size (MB)</th>
<th>Ratio All / Enc.</th>
<th>RTF (\times 10^{-2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sys. 1 (Tab. 1)</td>
<td>0</td>
<td>Baseline</td>
<td></td>
<td>32</td>
<td>7.2 / 7.0</td>
<td>15.9</td>
<td>12.7</td>
<td>15.3</td>
<td>10.4 / 10.3</td>
<td>16.5</td>
<td>286.3 / 217.1</td>
</tr>
<tr>
<td>Sys. 11 (Tab. 1)</td>
<td>1</td>
<td>Quant. Start. Point</td>
<td></td>
<td>32</td>
<td>7.4 / 7.1</td>
<td>15.4</td>
<td>12.7</td>
<td>15.4</td>
<td>10.5 / 10.3</td>
<td>16.5</td>
<td>286.3 / 217.1</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Quantized</td>
<td>Uniform</td>
<td>4</td>
<td>8.1</td>
<td>16.1</td>
<td>9.5</td>
<td>13.1</td>
<td>16.0</td>
<td>10.7</td>
<td>17.2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>Quantized</td>
<td>Manual</td>
<td>4</td>
<td>7.7</td>
<td>15.5</td>
<td>9.1</td>
<td>13.0</td>
<td>15.7</td>
<td>10.4</td>
<td>16.3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>Quantized</td>
<td>Deflate</td>
<td>8</td>
<td>10.4</td>
<td>15.1</td>
<td>8.2</td>
<td>13.0</td>
<td>15.7</td>
<td>10.4</td>
<td>16.3</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>Quantized</td>
<td>Mixed</td>
<td>2.6</td>
<td>7.9</td>
<td>18.9</td>
<td>9.2</td>
<td>13.3</td>
<td>15.9</td>
<td>10.8</td>
<td>17.4</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>Quant. Start. Point</td>
<td>KL</td>
<td>2.5</td>
<td>7.9</td>
<td>15.6</td>
<td>9.2</td>
<td>13.1</td>
<td>15.9</td>
<td>10.8</td>
<td>17.4</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>Quant. Start. Point</td>
<td>(Encoder Only)</td>
<td>4</td>
<td>7.4</td>
<td>15.4</td>
<td>9.0</td>
<td>12.9</td>
<td>15.6</td>
<td>10.4</td>
<td>16.8</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>Quant. Start. Point</td>
<td>(Encoder+Decoder)</td>
<td>4</td>
<td>7.3</td>
<td>15.4</td>
<td>9.0</td>
<td>12.9</td>
<td>15.6</td>
<td>10.4</td>
<td>16.8</td>
</tr>
</tbody>
</table>

The most compact and best performing model (Sys. 11) in Tab. 1 is selected as a streamlined starting point in the following low-bit quantization stage. During quantization, quantization-aware training (QAT) with the straight-through estimator (STE) [28] is adopted to reduce the performance degradation to quantization error. Results reported in Tab. 2 (ID 2-10) show several trends.

1) There is clear precision redundancy in un-quantized Conformer systems. For example, the uniformly configured 8-bit quantization of the Encoder incurs no significant WER increase over that of the 32-bit full-precision baseline (ID 4 vs. ID 0). 2) Systems with KL-configured mixed precision consistently outperform uniformly and manually configured ones of comparable averaged bit-widths: the 4-bit systems (ID 8 vs. ID 3) and the 2.5-bit systems (ID 6 vs. ID 5). The detailed local sub-layer bit-widths of 4-bit KL-configured mixed-precision (system ID 8) are shown in Fig. 2. 3) Compared with the full precision, manually configured baseline (Sys. 1, Tab. 1, also as ID 0 in Tab. 2), after also applying 4-bit quantization of the Decoder, an ultra compact system (ID 10) with an overall “lossless” compression ratio of 16.2 times (21.2 times for Encoder alone) was produced using both architecture compression and KL based mixed precision quantization while incurring no statistically significant WER increase. 4) The model architecture compression ratio is generally found to be linearly correlated with the changes of RTF (ID 0 vs. 1, Table 2 last column). However, the mixed-precision quantization methods are currently not fully supported by GPU CUDA libraries (e.g., 4-bit precision is still practically represented as 8-bit). Hence, the model quantization ratios have not been fully translated into the changes of RTF (ID 1 vs. 3, 4, Tab. 2).

7. Conclusion

This paper presents an aggressively compressed Conformer ASR system on the 300-hr Switchboard data using a novel combination of neural architectural compression and mixed-precision quantization approaches. They are designed to account for the varying performance sensitivity of different model components to structural compression and precision quantization. An overall “lossless” model compression ratio of 16.2 times (21.2 times for Encoder alone) is finally obtained over the full-precision manually configured baseline while incurring no statistically significant WER increase. Future work will focus on improving model compression and inference efficiency.

8. Acknowledgements

This research is supported by Hong Kong RGC GRF grant No. 14200220, 14200021, Innovation Technology Fund grant No. ITF/218/21.
9. References


