國立清華大學資訊工程系 114 學年度第一學期專題報告

專題名稱	ma-QAOA 對 BBV network 的 Max-cut 問題之改進				
參加競賽或計畫	☐ 參加對外競賽	☐ 參與其他計畫	☐無參加對外競賽或任何計畫		
學號	111062311				
姓名	林哲宇				

摘要

本專題研究提出改善 ma-QAOA 配 Adam 解 BBV network 的 Max-Cut 問題時遇到的收斂問題。我在 PennyLane lightning.gpu 模擬器上實作 ma-QAOA，並以 Pytorch 的 Adam 優化器進行參數訓練。因為 BBV network 的特性，其節點之加權度數 (即其與相臨之邊的權重總和) 具有兩極分布的特性。依據實驗發現，ma-QAOA 之 bata 參數的梯度訊號 (以 gradient variance 表示) 和其控制的節點之度數高度相關，可見該電路之梯度訊號也呈兩極分布，可能造成優化器收斂困難。針對該現象，我提出了兩種改良手法：階段參數分組與 Adam 動量重置，以避免梯度訊號不平衡造成的訓練困難並改善早期震盪 landscape 造成的後期收斂問題。

實驗針對節點數量為 12、15、18 的隨機 BBV network 各生成 20 個實例，每個實例進行 100 組隨機初始化參數實驗，統計其 approximation ratio (AR) 與收斂步數。結果顯示，參數分組與動量重置皆能顯著提升 AR，且同時使用時有疊加效果；在收斂速度方面，動量重置能大幅降低收斂步數，而參數分組在動量重置的情況下增加收斂步數，但在不動量重置的情況下減少收斂步數。本研究提供證據顯示，以 ma-QAOA 解 BBV network 的 Max-Cut 問題時，因該電路參數梯度訊號的兩極分布，可適用其他用來優化參數梯度訊號的兩極分布之模型的方法，如階段參數分組與 Adam 動量重置。

中華民國　　年　　月

I. 引言

隨著量子計算技術的發展，QAOA 被視為解決最佳化問題的重要候選方案。QAOA 能在固定深度的 VQC 上，透過經典優化器，逐步調整電路參數以逼近目標函數最大值。其中，Max-Cut 問題即為 QAOA 與其變種 ma-QAOA 可近似解決的熱門題目。

BBV network 為一圖類型，其有一特性為節點之加權度數 (即其與相臨之邊的權重總和) 具有兩極分布的特性，這導致解該類型圖 Max-Cut 問題之 ma-QAOA 電路出現梯度訊號不均的情況。文獻顯示經典優化器如 Adam 在梯度訊號不均衡的模型如 LLM 中表現可能受限，在訓練過程中，不同參數梯度尺度的落差會導致更新不平衡，進而影響收斂效率與最終性能。

本研究旨在探索 ma-QAOA 架構在解 BBV network 的 Max-Cut 問題之表現，並提出針對梯度訊號不均衡的 VQC 之訓練方法。我們提出參數分組與動量重置的方法，希望能提升 AR 與降低收斂步數，並透過經典模擬驗證其有效性。

II. 問題介紹

A. 最大割 (MAXCUT) 問題介紹

最大割為一圖學問題，即給定一圖$G(V,E)$，求一節點分割$\{V_0,V_1\}$，使所有跨越分割的邊權重和為最大，其數學表達式為 (1)。

$$\max_{x\in\{0,1\}^n}\sum_{1\le i<j\le|V|}w_{ij}(x_i\oplus x_j)=\sum_{(i,j,w_{ij})\in E}w_{ij}(x_i\oplus x_j) \tag{1}$$

其中x_i代表第i個節點所屬的分割集合 ($x_i=0$ iff $x_i\in V_0, x_i=1$ iff $x_i\in V_1$)，$x_i\oplus x_j=1$若且唯若兩節點分屬不同集合。

B. BBV network 介紹

BBV network 為一依照特定方法生成的圖類型，常常被用於模擬社交網絡等現實中網絡模型。給定m_0為初始核心節點數、w_0為新邊初始權重、m為新節點附著之舊節點數、N為總節點數，其生成算法為：

首先，定義節點v_i的「加權度數」$\deg_w(v_i)$為所有與其相連的邊之權重和，如 (2)。

$$\deg_w(v_i)=\sum_{j\in\mathcal{N}(i)}w_{ij} \tag{2}$$

其中$\mathcal{N}(i)$為所有與節點v_i相鄰的節點編號集合。

1. 準備初始狀態：
   生成包含m_0個節點的完全圖，將所有邊的權重都設為w_0。
2. 加入新節點：
   加入新節點v_{now}時，選擇不重複的m個既有節點與其附著。令U為既有節點之集合，選中v_i與新節點連接的機率函數為 (3)。

$$P(v_i)=\frac{\deg_w(v_i)}{\sum_{j\in U}\deg_w(v_j)} \tag{3}$$

為了避免重複選取，務實上的做法為建構一個大小為$|U|$的集合 pool 存放目前的既有節點編號：$\{0,1,\dots,|U|-1\}$，初始一個空集合 target，並重複進行以下操作m次：

a. 在 pool 中抽樣一個元素，選中i的機率為 (4)。

$$P(i) = \frac{\deg_w(v_i)}{\sum_{j\in \text{pool}} \deg_w(v_j)}, \forall i \in \text{pool} \tag{4}$$

b. 選中i後，將其加入 target，並刪除 pool 中的i。

如此一來，我們就得到了m個不重複的偏好連接點，即 target 中的節點，接著初始化所有新連接邊 (即$(v_{now}, v_i), \forall i \in \text{target}$) 之權重為$w_0$。此種連接方式讓原本加權度數較大的既有節點更有機會吸引新的節點，使其加權度數更大。

3. 更新邊權：

當新節點v_{now}與某個既有節點v_i相連時，依據 (5) 的規則更新所有與v_i相連的既有邊 (不包含新連接邊 (v_{now}, v_i)) 邊權，如圖 1。

$$w_{ij} \to w_{ij} + \alpha \cdot \frac{w_{ij}}{\deg_w(v_i)}, \forall j \in \mathcal{N}(i) \tag{5}$$

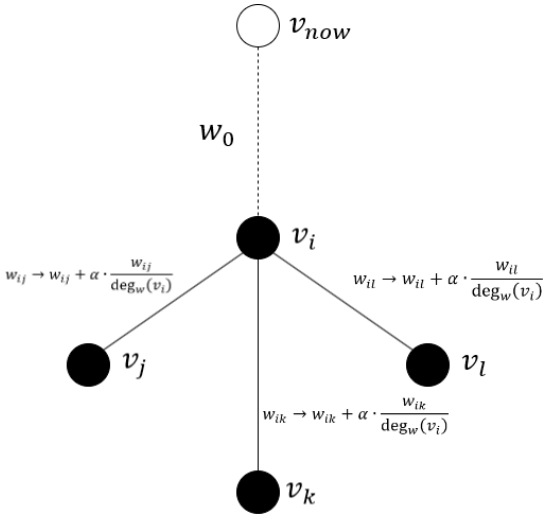

圖 1：新節點v_{now}與某個既有節點v_i相連時，既有邊權的更新情況
(實心為既有節點，空心為新增節點；實線為既有邊，虛線為新增邊)

此邊權更新方式相當於讓吸引到新節點的既有節點之加權度數更新如 (6)，讓其除了獲得新邊的w_0外，額外獲得α的獎勵，更加鞏固節點間加權度數大小的兩極化。

$$\deg_w(v_i) \to \deg_w(v_i) + w_0 + \alpha \tag{6}$$

4. 補完節點數：

重複 2.直到節點數為N。

因為其獨特的生成機制，造成節點的帶權度數呈兩極分布，即大部分節點不是有極大的帶權度數，就是有極小的帶權度數，有著明顯的「核心」，如圖 2。[]

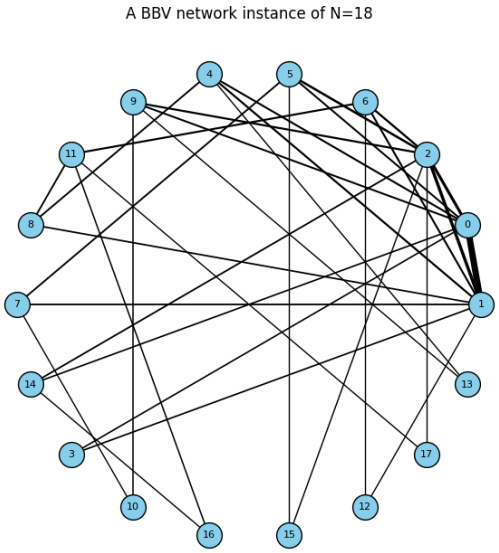

圖 2：$|V| = 18$時，一 BBV network 隨機實例，其中邊的粗細正比於權重

III. 量子電路架構介紹

A. QAOA 架構介紹

QAOA 為一用於解最佳化問題的分變量子電路 (VQC)，其目標為給定一定義域$x \in \{0,1\}^n$，值域為$\mathbb{R}$的目標函數$f(x)$，找一x使$f(x)$盡可能的大。

單層的 QAOA 量子電路包含兩個參數γ及β與兩個由 Hamiltonian 矩陣與參數控制的 Unitary operator：$U_p(\gamma) = e^{-i\gamma H_p}$以及$U_M(\beta) = e^{-i\beta H_M}$。其中$H_p$為一和目標函數相關的對角矩陣，定義為將每個 computation basis projector 乘上該 computation basis 的目標函數值之和，如 (7)。

$$H_p = \sum_{x\in\{0,1\}^n} f(x)|x\rangle\langle x| \tag{7}$$

H_M的定義則為 N 個不同的單位元 Pauli X 矩陣的和，如 (8)。

$$H_M = \sum_{1\le i\le n} \sigma_i^x \tag{8}$$

透過數值方法調整γ及β，讓每個$|x\rangle$因$U_p(\gamma)$獲得一個相位後，再透過$U_M(\beta)$將狀態之間引入疊加，讓每個有不同相位的$|x\rangle$帶著相位分散至位元相鄰的 computational basis，形成建設性和破壞性干涉，進而增強目標$|x\rangle$的振幅，削弱非目標$|x\rangle$的振幅。另外，一般 QAOA 的初始狀態會選用均勻疊加態，即$|+ + \cdots +\rangle$，以確保初始不偏袒任何 computational basis，見圖 3。

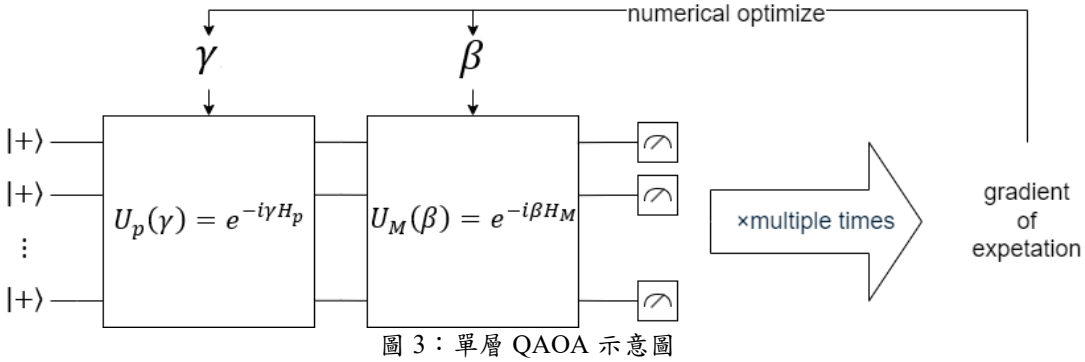

圖 3：單層 QAOA 示意圖

p層的 QAOA 電路即為p層U_p及U_M的串聯，每層採用不同的γ及β，因此參數為兩個長度為p的陣列$\{\gamma_1, \gamma_2, \dots, \gamma_p\}$及$\{\beta_1, \beta_2, \dots, \beta_p\}$，透過多層的架構，能在淺層結果的基礎上，繼續增強目標$|x\rangle$的振幅，削弱非目標$|x\rangle$的振幅，以得到更好的結果。[2]

B. ma-QAOA 架構介紹

ma-QAOA 為 QAOA 的變種，在原 QAOA 中，$U_p(\gamma) = e^{-i\gamma H_p}$，$U_M(\beta) = e^{-i\beta H_M}$，若問題可拆成$k$個相加項，如 (9)。

$$f(x) = \sum_{1 \le i \le k} c_i(x) \tag{9}$$

則可將$U_p(\gamma)$拆成k個互相可對易的部分，如 (10)。

$$U_p(\gamma) = e^{-i\gamma H_p} = e^{-i\gamma \sum_{x\in\{0,1\}^n} f(x)|x\rangle\langle x|} = e^{-i\gamma \sum_{x\in\{0,1\}^n} \sum_{1\le i\le k} C_i(x)|x\rangle\langle x|}$$
$$= \prod_{1 \le i \le k} e^{-i\gamma \sum_{x\in\{0,1\}^n} c_i(x)|x\rangle\langle x|} \tag{10}$$

因H_M為n個不同的單位元σ^x的和，可將$U_M(\beta)$拆成n個互相可對易的部分，如 (11)。

$$U_M(\beta) = e^{-i\beta H_M} = e^{-i\beta \sum_{1\le i\le n} \sigma^x{}_i} = \prod_{1 \le i \le n} e^{-i\beta \sigma^x{}_i} \tag{11}$$

若將$U_p(\gamma)$及$U_M(\beta)$的每個不同的可對易單元賦予獨立的γ及β，即$U_p(\gamma)$改寫成$U_p(\gamma_1, \gamma_2, \dots, \gamma_k)$，$U_M(\beta)$改寫成$U_M(\beta_1, \beta_2, \dots, \beta_n)$，如 (12)。

$$U_p(\gamma_1, \gamma_2, \dots, \gamma_k) = \prod_{1 \le i \le k} e^{-i\gamma_i \sum_{x\in\{0,1\}^n} c_i(x)|x\rangle\langle x|}$$
$$U_M(\beta_1, \beta_2, \dots, \beta_n) = \prod_{1 \le i \le n} e^{-i\beta_i \sigma^x{}_i} \tag{12}$$

此即為 ma-QAOA 架構，該模型可透過較多的參數自由度，進而達到比 QAOA 架構更好的結果，見圖 4。[3]

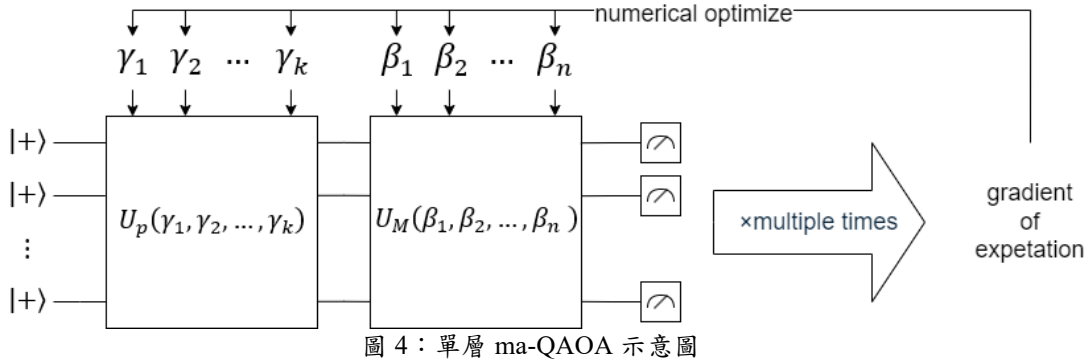

圖 4：單層 ma-QAOA 示意圖

IV. 數值優化方法介紹

A. Adaptive Moment Estimation (Adam) 介紹

Adam 是一種基於目標函數梯度的數值優化方法，給定α為初始學習率，β_1為一階動量衰減率，β_2為二階動量衰減率，m_t為一階動量，v_t為二階動量，$\hat{m}_t$為修正後的一階動量，$\hat{v}_t$為修正後的二階動量，θ_t為參數，g_t為目標函數的梯度，ϵ為學習率修正量 (值得注意的是$g_t{}^2$代表g_t的逐元素平方，$\sqrt{\hat{v}_t}$代表$\hat{v}_t$的逐元素根號)，參數更新方法如 (13)。

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) {g_t}^2$$
$$\hat{m}_t = \frac{m_t}{1-{\beta_1}^t}, \qquad \hat{v}_t = \frac{v_t}{1-{\beta_2}^t}$$
$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{13}$$

相較於原始的梯度下降法，Adam 引入了兩個新的特性：

1. 動量法 (momentum)：

   透過$\hat{m}_t$更新而非g_t，讓歷史梯度的加權平均決定更新方向而非當前梯度，這樣的設定保障了遇到粗糙地形時的越障能力。

2. 自適應學習率 (adaptive learning rate)：

   每個參數的更新量${\hat{m}_t}^{(i)}$都由歷史的梯度方均根$\sqrt{{\hat{v}_t}^{(i)}}$進行 nornalize，讓梯度大的參數步伐變小，梯度小的參數步伐變大，讓所有參數都能以比較均衡的速度學習。在單一參數梯度變化大時，也能在梯度大時縮小學習率，梯度小時加大學習率。

V. 完整方法與我的改良

A. 基本設定

關於數值優化器，我使用 torch 中的 Adam 優化器，採用預設的β_1及β_2，α設為 0.01。關於量子電路模擬器，我使用 pennylane 的 lightning.gpu，並設定 shots=None，即直接用 statevector 取期望值，不做抽樣，偏微分方法的部分則選擇最有效率且穩定的 adjoint 方法。關於 QAOA 架構，我都以層數為 1 進行實驗。

B. 單純方法

我的目標是「用 ma-QAOA 的方法解 BBV network 的 MaxCut 問題」，因 MaxCut 問題可拆成$|E|$個相加項，並且可用$x \in \{0,1\}^{|V|}$代表節點所屬的分割集合，我們設計：

$$U_p(\gamma_1, \gamma_2, \dots, \gamma_{|E|}) = \prod_{(i,j,w_{ij}) \in E} e^{-i\gamma_i w_{ij}(x_i \oplus x_j)}$$
$$= \prod_{(i,j,w_{ij}) \in E} \mathrm{CNOT}(i,j) \cdot R_Z(2\gamma_i w_{ij}) \cdot \mathrm{CNOT}(i,j)$$
$$U_M(\beta_1, \beta_2, \dots, \beta_{|V|}) = \prod_{1 \le i \le |V|} e^{-i\beta_i {\sigma^x}_i} = \prod_{1 \le i \le |V|} R_X(\beta_i) \tag{14}$$

接著我直接使用數值優化器來優化所有的γ及β，直到收斂。

我將我採用的收斂判斷方法稱為「差值收斂」，即在 10 次更新內，loss 值的誤差小於總邊權的10^{-6}，如 (15)，則判定為收斂。

$$\max(loss_{t-9:t}) - \min(loss_{t-9:t}) < 10^{-6} \cdot \sum_e w_e \tag{15}$$

C. 可能的改良方向

考慮到 BBV network「節點的帶權度數呈兩極分布」，我觀察到β參數的梯度變異數和其控制的節點之帶權度數統計高度相關，見圖 5 與表 1。

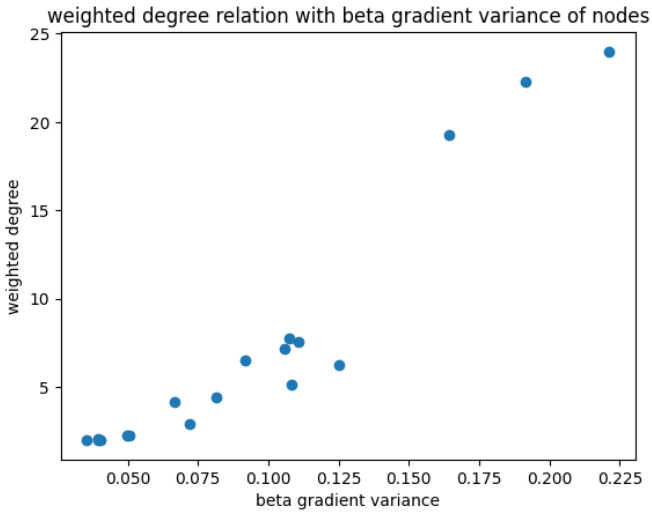

圖 5：$|V| = 18$時，一隨機實例的 beat gradient variance 和 weighted degree 分布情況

表 1：$|V| = 12$、$|V| = 15$及$|V| = 18$之隨機 BBV network 的
beat gradient variance 和 weighted degree 之平均相關係數 (四捨五入至小數點後第 6 位)

$\|V\| = 12$	$\|V\| = 15$	$\|V\| = 18$
0.969793	0.968220	0.958407

在介紹我的優化方法前，先介紹另一收斂判斷方法，我稱為「斜率收斂」，即若最近 10 次更新的平均斜率小於從階段開始到 10 次前的斜率的 1/2，則判定為收斂，如 (16)。

$$\frac{loss_{t-9} - loss_t}{10 - 1} < \frac{1}{2} \cdot \frac{loss_1 - loss_t}{|loss| - 1} \tag{16}$$

值得注意的是，該方法並非用來判斷「完全收斂」，而是「優化效果漸弱」，適合進入下一階段。我用該方法來判斷是否進入下一階段而非「差值收斂」是為了避免在非最終階段浪費過多步數來精細優化。

針對於優化 AR 與收斂步數，我提出了兩種優化方法：

1. 階段參數分組：

   將節點依加權度數高到低排序，並分成三組。訓練分為三個階段，第一階段只訓練和第一組節點有關的參數 (該點的β和該點臨邊的γ)；第一階段訓練和前二組有關的參數；第三階段訓練所有參數。此舉是為了讓優化器先處理梯度訊號較大的參數，避免因考慮梯度訊號較小的參數而拖垮表現，以提升效果。值得注意的是，因為早期的參數凍結，可能導致收斂速度變慢，因此我提出依照參數控制之節點的加權度數加大非最終階段的學習率，如 (17)，希望透過學習率補償來減緩該副作用。

   $$\alpha_T = \alpha \cdot \frac{\sum_{v_i \in V} \deg_w(v_i)}{\sum_{v_i \in S(T)} \deg_w(v_i)} \tag{17}$$

   其中α_T為階段T的學習率，α為設定學習率 0.01，$S(T)$為階段T可訓練的參數所控制之節點集合，即加權度數高到低排序前 T 組的節點聯集。

2. 階段重置 Adam 動量：

   在每個階段開始時，清除過去的 Adam 動量資訊。此舉是為了避免在非梯度谷地區域的梯度資訊影響梯度谷地地區的收斂，提升效果與收斂速度。[]

以上兩者皆共三個階段，用「斜率收斂」判斷是否進入下一階段，用「差值收斂」判斷第三段收斂。

VI. 實驗結果

我的實驗對象為四種算法：單純 Adam、Adam+動量重置、Adam+參數分組、Adam+動量重置+參數分組。對於以上四種算法，我進行了其對$|V| = 12$、$|V| = 15$及$|V| = 18$之隨機生成 BBV network 的實驗。每個節點數量的情況包含 20 個圖實例，每個實例用$\frac{\pi}{8}$角度隨機初始化[5]生成 100 組隨機的初始γ及初始β，並統計其 AR 與收斂步數的分佈。

A. Approximation ratio 分析

根據實驗結果統計的圖 6、圖 7、圖 8、表 2，可以看到四種算法的 AR 大約：Adam+動量重置+參數分組> Adam+參數分組≅Adam+動量重置>單純 Adam，可見參數分組與動量重置都有助於 AR 的提升，並且一起使用有疊加效果。

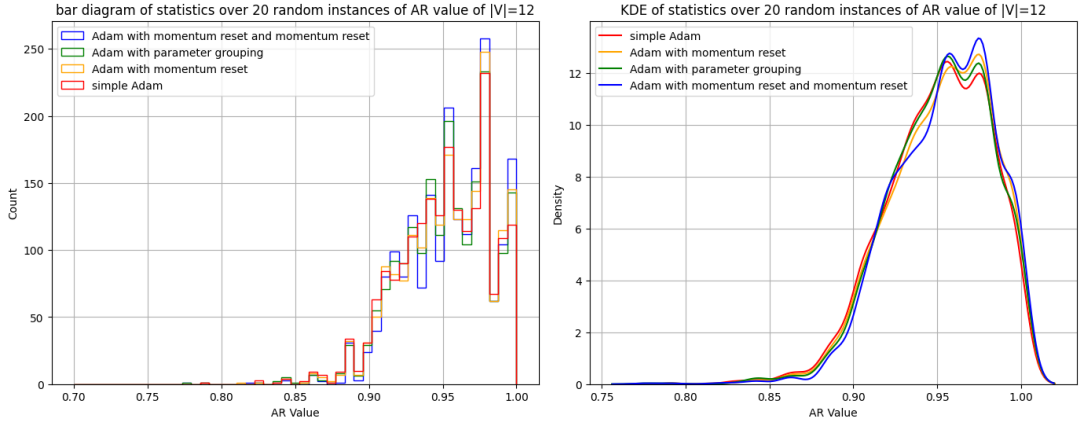

圖 6：$|V| = 12$時，20 個隨機 BBV network 實例，每個實例取 100 次隨機初始參數的 AR 之分布直方圖與 KDE 統計圖

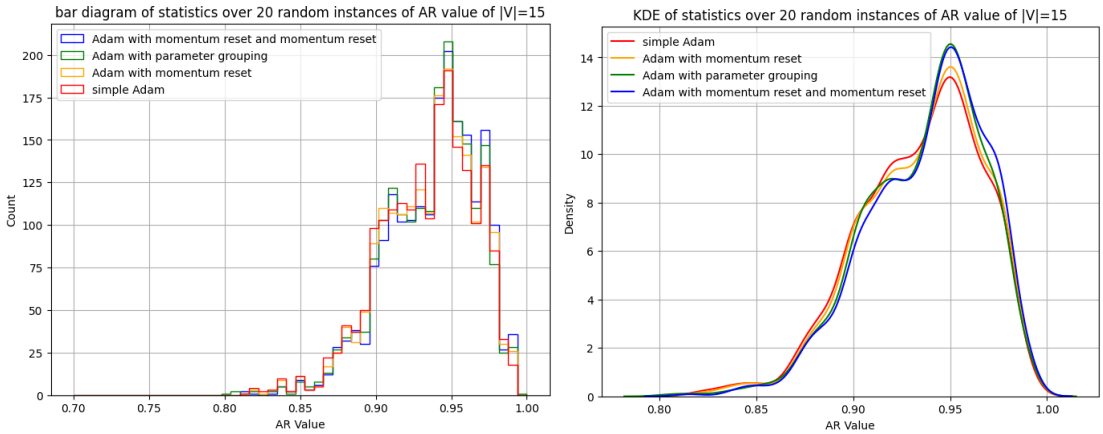

圖 7：$|V| = 15$時，20 個隨機 BBV network 實例，每個實例取 100 次隨機初始參數的 AR 之分布直方圖與 KDE 統計圖

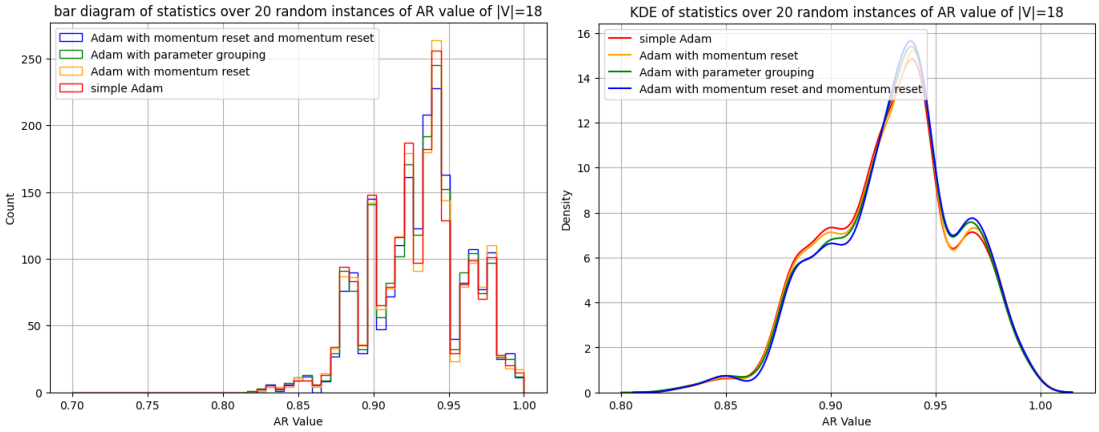

圖 8：$|V| = 18$時，20 個隨機 BBV network 實例，每個實例取 100 次隨機初始參數的 AR 之分布直方圖與 KDE 統計圖

表 2：$|V| = 12$、$|V| = 15$及$|V| = 18$時，20 個隨機 BBV network 實例，每個實例取 100 次隨機初始參數的 AR 之平均值 (四捨五入至小數點後第 6 位)

	單純 Adam	Adam+動量重置	Adam+參數分組	Adam+動量重置+參數分組
$\|V\| = 12$	0.950650	0.952539	0.951929	0.954123
$\|V\| = 15$	0.934376	0.935702	0.936555	0.938337
$\|V\| = 18$	0.929266	0.930023	0.930319	0.931284

B. 收斂步數分析

根據實驗結果統計的圖 9、圖 10、圖 11、表 3，可以看到四種算法的收斂步數大約：Adam+動量重置<Adam+動量重置+參數分組≪Adam+參數分組<單純 Adam， 可以注意到動量重置顯著降低了收斂步數，而參數分組在動量重置的情況下，因為早期參數凍結的關係，速度落後於不分組凍結只動量重置的算法；在不動量重置的情況下，也是因為早期參數凍結的關係，讓 Adam 優化器少累積了凍結參數的高震盪動量 (這正是動量重置可避免的，因此在動量重置的情況下，參數凍結並無此優勢)，使得收斂步數優於不分組凍結也不動量重置的算法。

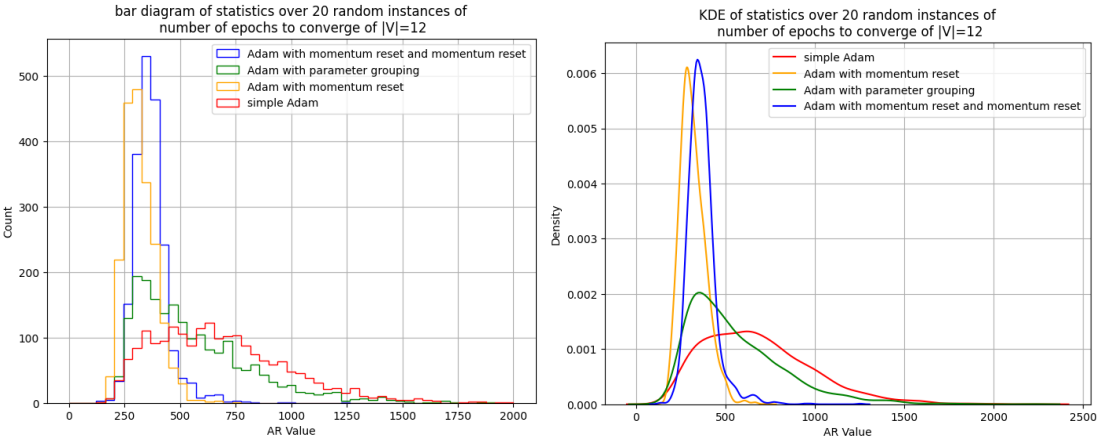

圖 9：$|V| = 12$時，20 個隨機 BBV network 實例，每個實例取 100 次隨機初始參數的收斂步數之核密度估計 KDE

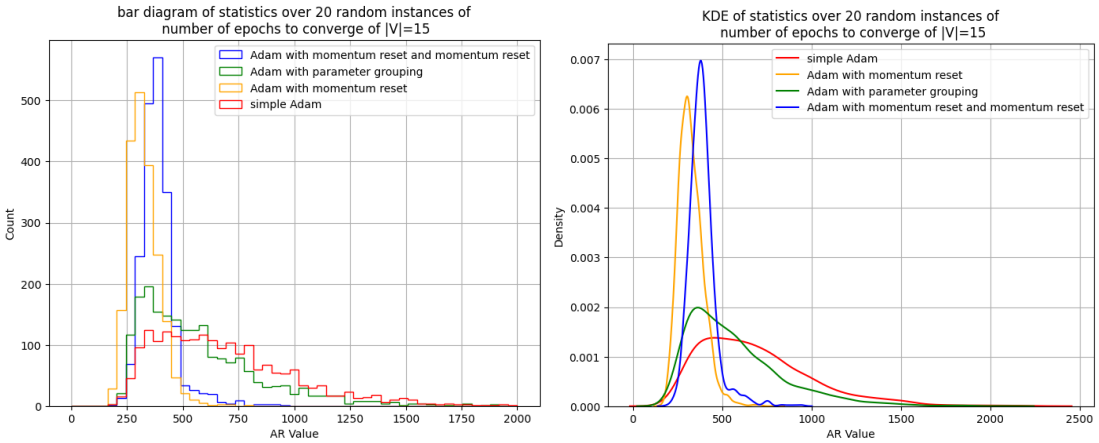

圖 10：$|V| = 15$時，20 個隨機 BBV network 實例，每個實例取 100 次隨機初始參數的收斂步數之核密度估計 KDE

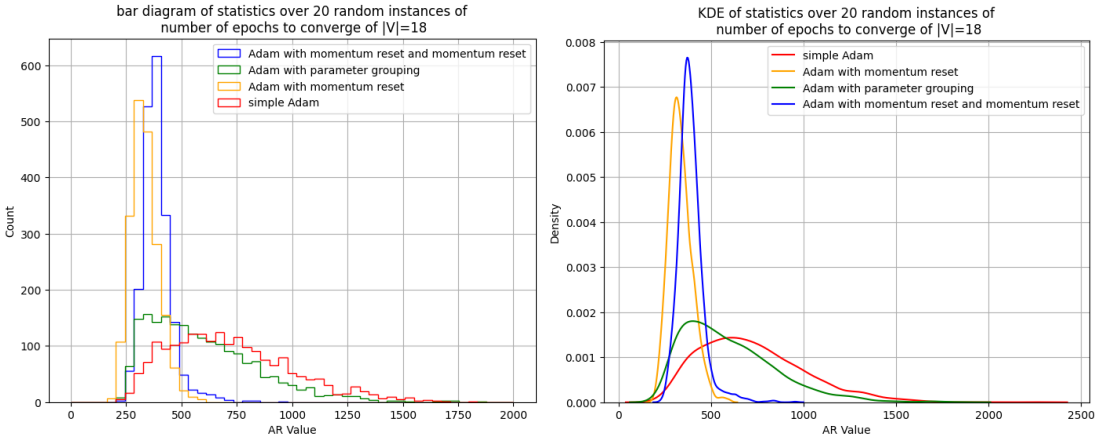

圖 11：$|V| = 18$時，20 個隨機 BBV network 實例，每個實例取 100 次隨機初始參數的收斂步數之核密度估計 KDE

表 2：$|V| = 12$、$|V| = 15$及$|V| = 18$時，20 個隨機 BBV network 實例，每個實例取 100 次隨機初始參數的 AR 之平均值 (四捨五入至小數點後第 6 位)

	單純 Adam	Adam+動量重置	Adam+參數分組	Adam+動量重置+參數分組
$\|V\| = 12$	681.3695	319.33	556.1095	368.565
$\|V\| = 15$	686.8945	324.782	567.623	388.296
$\|V\| = 18$	721.0615	334.479	597.096	389.199