

專題名稱	用 ma-QAOA 搭配 Adam 解 BBV 網絡的 Max-Cut 問題之方法改良				
參加競賽或計畫	<input type="checkbox"/> 參加對外競賽	<input type="checkbox"/> 參與其他計畫		<input type="checkbox"/> 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 (Barrat–Barthelemy–Vespignani) 網絡的 Max-Cut 問題時遇到的特定問題。因為 BBV 網絡的特性，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，可見該電路參數之梯度訊號也呈兩極分布。根據文獻顯示，參數之梯度訊號分布不均可能造成數值優化器收斂困難。針對該現象，我提出了兩種改良手法：**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。

我在 PennyLane lightning.gpu 模擬器上選擇 shots=None (即期望值直接靠 statevector 統計模擬，而非照機率抽樣) 的裝置實作 ma-QAOA，並以 Pytorch 的 Adam 優化器對 ma-QAOA 輸出之期望值進行參數訓練。針對節點數量為 12、15、18 的隨機 BBV 網絡，各生成 20 個實例，每個實例進行 100 組隨機初始化參數，統計其 approximation ratio 與收斂步數。結果顯示，參數分組與動量重置皆能顯著提升 approximation ratio，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，用 ma-QAOA 搭配 Adam 解 BBV 網絡的 Max-Cut 問題時，因該電路參數梯度訊號的兩極分布，階段參數分組與 Adam 動量重置可提升表現與效率。

I. 引言

隨著近期量子計算硬體的進展，變分量子電路 (Variational Quantum Circuit, VQC) 逐漸成為混合式量子-經典演算法的核心框架。VQC 透過設計含有可調參數的量子電路，並依賴經典優化器反覆更新這些參數，以期最小化或最大化某一目標函數。此方法在解決量子化學模擬、機器學習以及組合最佳化問題等領域展現出潛在優勢，因此被廣泛視為含噪聲中尺度量子 (NISQ) 時代中的關鍵工具。由於其所需電路深度相對較短，能在現有硬體上展現更強的噪聲耐力，這使得 VQC 成為連結當前量子裝置與實際應用的重要橋樑。

在眾多基於 VQC 的演算法中，量子近似最佳化演算法 (Quantum Approximate Optimization Algorithm, QAOA) 是最具代表性的模型之一。QAOA 藉由交替施加問題哈密頓量 (Phase Hamiltonian) 與混合哈密頓量 (Mixer Hamiltonian)，並透過經典優化器調整參數，逐步放大期望解的振幅，以近似求解組合最佳化問題。

在 QAOA 的基礎上，多角度 QAOA (multi-angle QAOA, ma-QAOA) 進一步增加了電路的表達能力。傳統 QAOA 在每一層僅使用兩個全域參數 (γ 與 β)，分別控制問題哈密頓量與混合哈密頓量的演化；相較之下，ma-QAOA 為每一項問題哈密頓量與每一個 qubit 的混合操作引入獨立的參數，使得電路在同樣深度下擁有更多自由度。這種設計能更靈活地捕捉問題結構，提升近似最佳化的表現，但同時也帶來了更大的參數空間與訓練挑戰。

最大割 (Max-Cut) 問題是 QAOA 與 ma-QAOA 最常用來驗證效能的基準案例之一。Max-Cut 問題屬於 NP-hard 問題，其目標是在給定圖 $G(V, E)$ 上，將節點集合 V 劃分為兩個子集，使跨越兩個子集的邊權重和最大化。Max-Cut 不僅在理論計算領域中具有挑戰性，也在實務應用上具有廣泛意義，例如社群網絡分析、VLSI 設計與金融投資組合等，因此成為 QAOA 及其變種演算法的經典測試平台。

在 VQC 的訓練中，經典數值最佳化方法扮演關鍵角色。其中，Adam (Adaptive Moment Estimation) 作為當前最常用的優化器之一，因其結合動量 (momentum) 與自適應學習率 (adaptive learning rate) 的特性，通常能在複雜函數空間中提供穩定且快速的收斂。

Barrat-Barthelemy-Vespignani (BBV) 網絡因能有效模擬現實世界中的複雜網絡 (如社交網絡、資訊傳播網絡等) 而受到關注。其生成機制不僅考慮節點間的偏好連接 (preferential attachment)，還會隨著新節點的加入而動態調整既有邊的權重。這種設計使得網絡的節點加權度數 (weighted degree，即與該節點相鄰邊權重的總和) 呈現高度兩極化分布：少數「核心節點」累積了極大的加權度數，而大部分節點則維持相對較低的加權度數。

這樣的結構特性雖能貼近真實網絡中「強者愈強」的動態，但同時也對 QAOA 與 ma-QAOA 的訓練造成挑戰。由於不同節點所對應的參數梯度幅度差異懸殊，電路在優化過程中可能出現梯度不均衡 (gradient imbalance) 現象。當經典優化器 (例如 Adam) 面對此類兩極化的梯度訊號時，往往難以同時兼顧所有參數的更新步調，進而導致收斂效率下降甚至影響最終解的品質。因此，如何針對 BBV 網絡的結構特性設計更合適的優化策略，成為提升 ma-QAOA 效能的關鍵課題。

基於此觀察，本研究聚焦於改善 ma-QAOA 於 BBV 網絡 Max-Cut 問題中的收斂行為。我們提出兩項方法：(1) 階段式參數分組，透過分批訓練不同梯度尺度的參數以緩解更新不均衡；(2) Adam 動量重置，在訓練階段切換時清除累積的歷史動量，以避免早期高度震盪的梯度資訊延續至後期。透過經典模擬驗證，本研究展示了這些策略在提升近似比（Approximation Ratio, AR）與降低收斂步數方面的有效性，並為未來針對具有梯度兩極化特性的 VQC 訓練問題提供了新的研究方向。

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)。

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

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 網絡介紹

BBV 網絡為一依照特定方法生成的圖類型。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、 α 為邊權更新參數、 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)。

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

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. 加入新節點與更新邊權：

加入新節點 v_{now} 時，選擇不重複的 m 個既有節點與其附著，並更新被附著點之臨邊權重。具體方法為：

初始一個空集合 $chosen$ ，並重複進行以下操作 m 次：

- 令 S 為既有節點之集合， $V = S - chosen$ 為「不在 $chosen$ 中的既有節點之集合」。在集合 V 中抽樣一個節點，選中 v_i 之機率為 (3)。

$$P(i) = \frac{\deg_w(v_i)}{\sum_{j \in V} \deg_w(v_j)}, \forall i \in V \quad (3)$$

- 依據 (4) 的規則更新所有與 v_i 相連的既有邊之權重（不包含新連接邊 (v_{now}, v_i) ），如圖 1。

$$w_{ij} \rightarrow w_{ij} + \alpha \cdot \frac{w_{ij}}{\deg_w(v_i)}, \forall j \in \mathcal{N}(i) \quad (4)$$

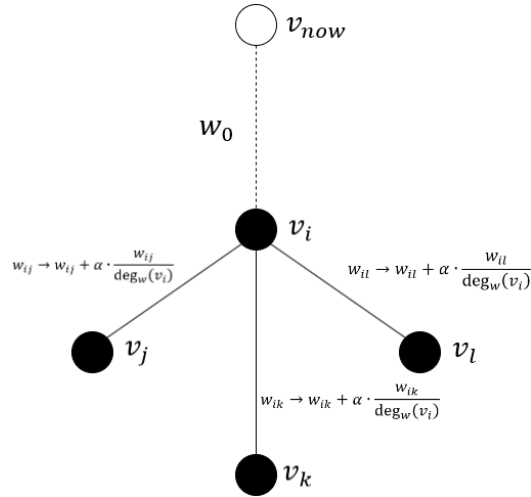


圖 1：新節點 v_{now} 與某個既有節點 v_i 相連時，既有邊權的更新情況
(實心為既有節點，空心為新增節點；實線為既有邊，虛線為新增邊)

3. 重複步驟 2. 直到圖有 N 個節點。

2.a. 的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；而 2.b. 的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 (5)，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。

$$\deg_w(v_i) \rightarrow \deg_w(v_i) + w_0 + \alpha \quad (5)$$

因為其獨特的生成機制，可見得 BBV 網絡節點之加權度數具有兩極分布的特性。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。[1]

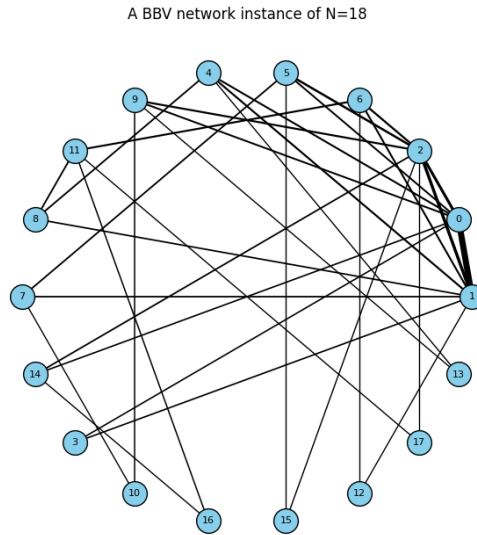


圖 2： $|V| = 18$ 時，一 BBV 網絡隨機實例，其中邊的粗細正比於權重

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 (6)。

$$H_p = \sum_{x \in \{0,1\}^n} f(x) |x\rangle\langle x| \quad (6)$$

H_M 的定義則為 N 個不同的單位元 Pauli X 矩陣的和，如 (7)。

$$H_M = \sum_{1 \leq i \leq n} \sigma_i^x \quad (7)$$

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 $|++\dots+\rangle$ ，以確保初始不偏袒任何 computational basis，見圖 3。

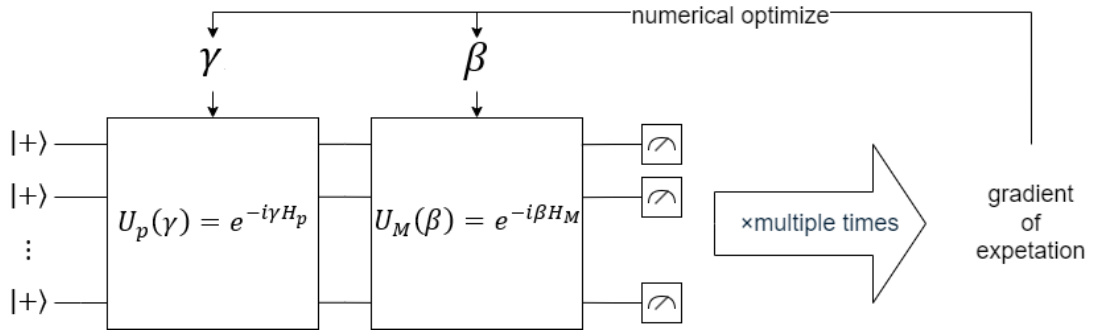


圖 3：單層 QAOA 示意圖

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 (8)。

$$f(x) = \sum_{1 \leq i \leq k} c_i(x) \quad (8)$$

則可將 $U_p(\gamma)$ 拆成 k 個互相可對易的部分，如 (9)。

$$\begin{aligned} U_p(\gamma) &= e^{-i\gamma H_p} = e^{-i\gamma \sum_{x \in \{0,1\}^n} f(x) |x\rangle\langle x|} = e^{-i\gamma \sum_{x \in \{0,1\}^n} \sum_{1 \leq i \leq k} c_i(x) |x\rangle\langle x|} \\ &= \prod_{1 \leq i \leq k} e^{-i\gamma \sum_{x \in \{0,1\}^n} c_i(x) |x\rangle\langle x|} \end{aligned} \quad (9)$$

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 (10)。

$$U_M(\beta) = e^{-i\beta H_M} = e^{-i\beta \sum_{1 \leq i \leq n} \sigma_i^x} = \prod_{1 \leq i \leq n} e^{-i\beta \sigma_i^x} \quad (10)$$

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 (11)。

$$U_p(\gamma_1, \gamma_2, \dots, \gamma_k) = \prod_{1 \leq i \leq k} e^{-i\gamma_i \sum_{x \in \{0,1\}^n} c_i(x) |x\rangle\langle x|}$$

$$U_M(\beta_1, \beta_2, \dots, \beta_n) = \prod_{1 \leq i \leq n} e^{-i\beta_i \sigma_i^x} \quad (11)$$

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]

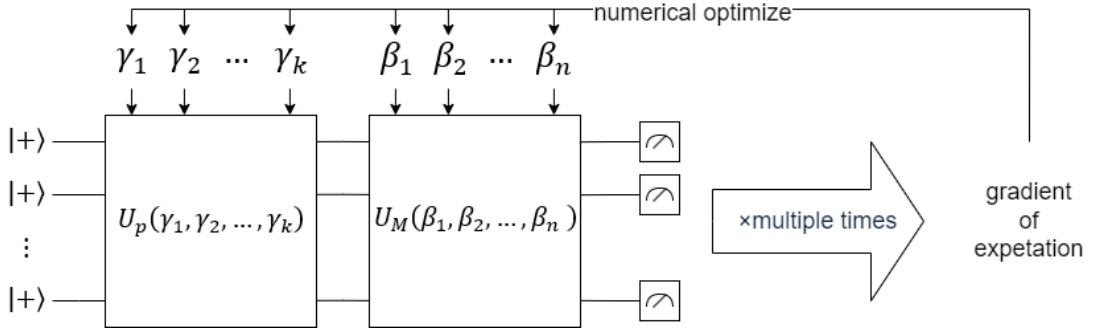


圖 4：單層 ma-QAOA 示意圖

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 (12)。

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \\ \theta_{t+1} &= \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \end{aligned} \quad (12)$$

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 normalize，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 $\beta_1 = 0.9$ 及 $\beta_2 = 0.999$ ， 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。[\[4\]](#)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 網絡的 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：

$$\begin{aligned} U_p(\gamma_1, \gamma_2, \dots, \gamma_{|E|}) &= \prod_{(i,j,w_{ij}) \in E} e^{-i\gamma_i w_{ij}(x_i \oplus x_j)} \\ &= \prod_{(i,j,w_{ij}) \in E} \text{CNOT}(i,j) \cdot R_z(2\gamma_i w_{ij}) \cdot \text{CNOT}(i,j) \\ U_M(\beta_1, \beta_2, \dots, \beta_{|V|}) &= \prod_{1 \leq i \leq |V|} e^{-i\beta_i \sigma^x_i} = \prod_{1 \leq i \leq |V|} R_x(\beta_i) \end{aligned} \quad (13)$$

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 (14)，則判定為收斂。

$$\max(\text{loss}_{t-9:t}) - \min(\text{loss}_{t-9:t}) < 10^{-6} \cdot \sum_e w_e \quad (14)$$

C. 我的改良方向

考慮到 BBV 網絡「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。

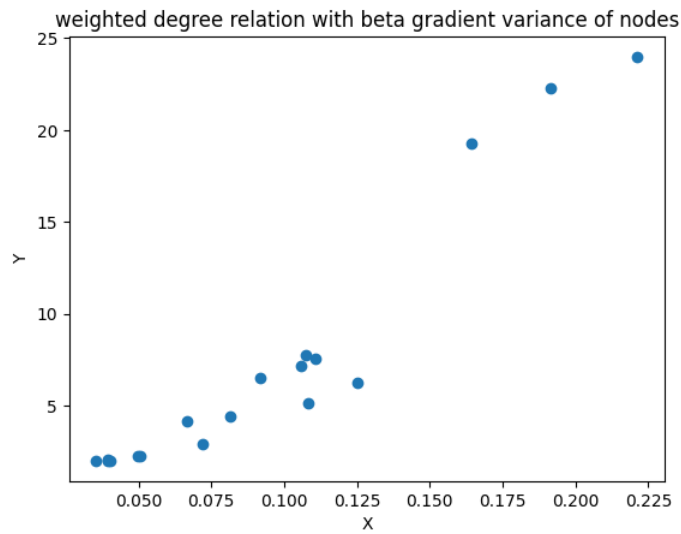


圖 5： $|V| = 18$ 時，一隨機實例的 beat gradient variance 和 weighted degree 分布情況

表 1： $|V| = 12$ 、 $|V| = 15$ 及 $|V| = 18$ 之隨機 BBV 網絡的
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$
0.969600	0.968065	0.958095

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 (15)。

$$\frac{loss_{t-9} - loss_t}{10 - 1} < \frac{1}{2} \cdot \frac{loss_1 - loss_t}{|loss| - 1} \quad (15)$$

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 (16)，希望透過學習率補償來減緩該副作用。

$$\alpha_T = \alpha \cdot \frac{\sum_{v_i \in V} \deg_w(v_i)}{\sum_{v_i \in S(T)} \deg_w(v_i)} \quad (16)$$

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。[\[5\]](#)

以上兩者皆分三個階段，用「斜率收斂」判斷是否進入下一階段，用「差值收斂」判斷吹後段收斂。

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 網絡的實驗。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[\[6\]](#)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 \cong 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。

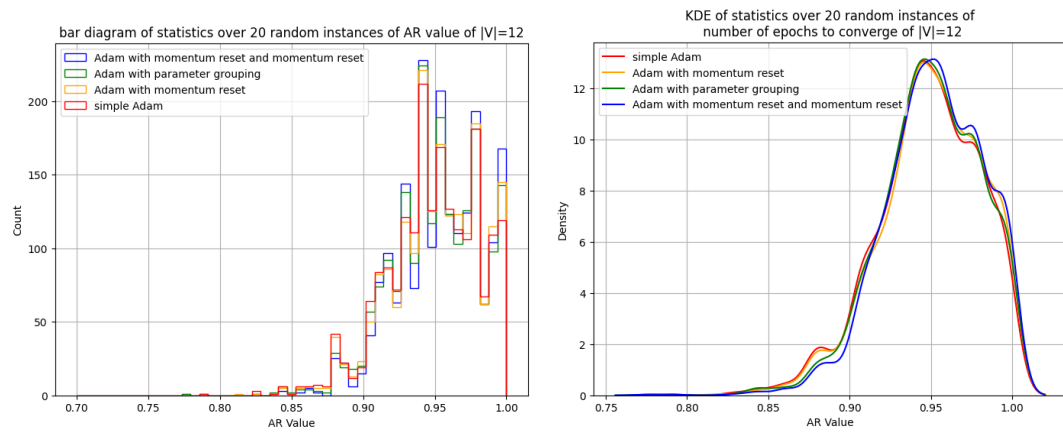


圖 6： $|V| = 12$ 時，20 個隨機 BBV 網絡實例，每個實例取 100 次隨機初始參數的 AR 之分布直方圖與核密度估計 (KDE) 圖

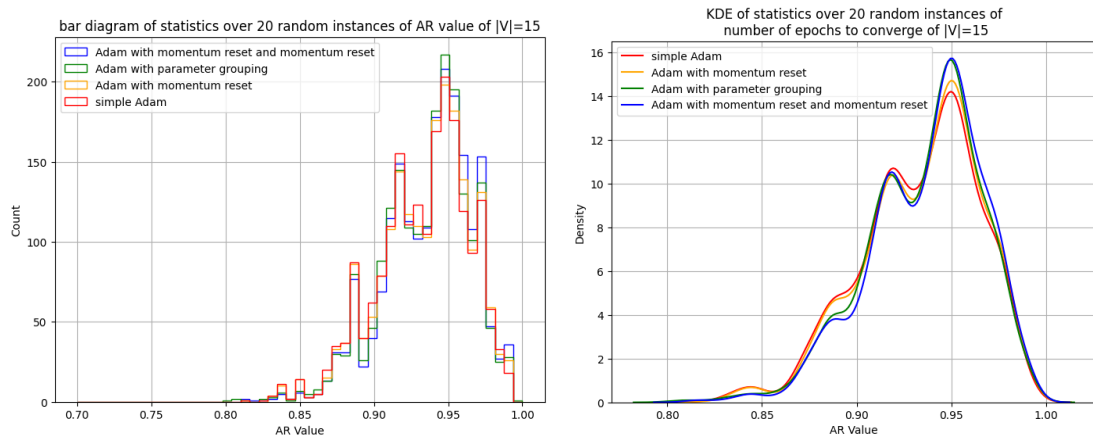


圖 7： $|V| = 15$ 時，20 個隨機 BBV 網絡實例，每個實例取 100 次隨機初始參數的 AR 之分布直方圖與核密度估計 (KDE) 圖

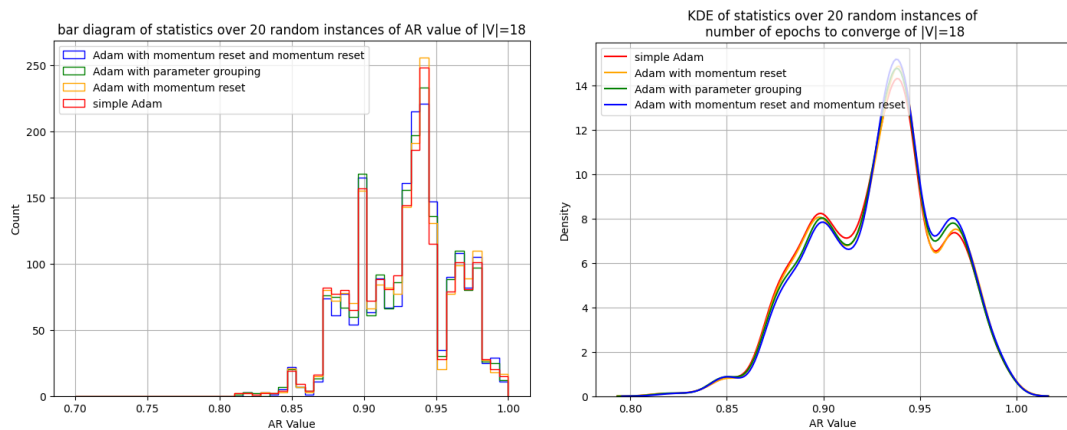


圖 8： $|V| = 18$ 時，20 個隨機 BBV 網絡實例，每個實例取 100 次隨機初始參數的 AR 之分布直方圖與核密度估計 (KDE) 圖

表 2： $|V| = 12$ 、 $|V| = 15$ 及 $|V| = 18$ 時，20 個隨機 BBV 網絡實例，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$	0.948438	0.950095	0.949901	0.952151
$ V = 15$	0.933290	0.934548	0.935386	0.937142
$ V = 18$	0.927683	0.928486	0.928667	0.929691

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。

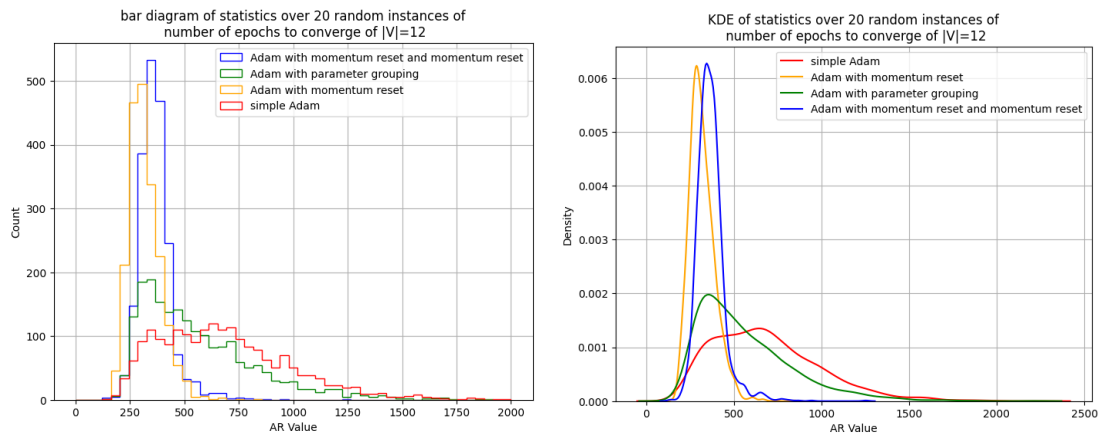


圖 9： $|V| = 12$ 時，20 個隨機 BBV 網絡實例，每個實例取 100 次隨機初始參數的收斂步數之分布直方圖與核密度估計 (KDE) 圖

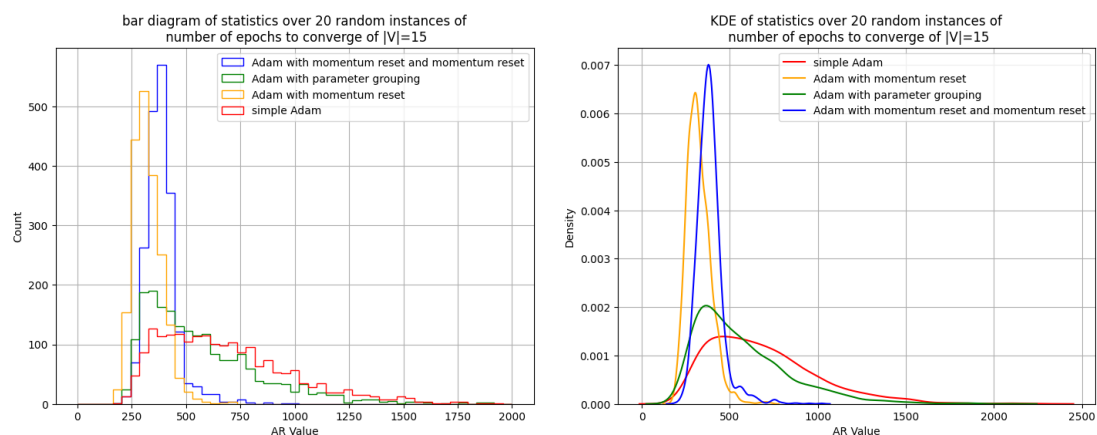


圖 10： $|V| = 15$ 時，20 個隨機 BBV 網絡實例，每個實例取 100 次隨機初始參數的收斂步數之分布直方圖與核密度估計 (KDE) 圖

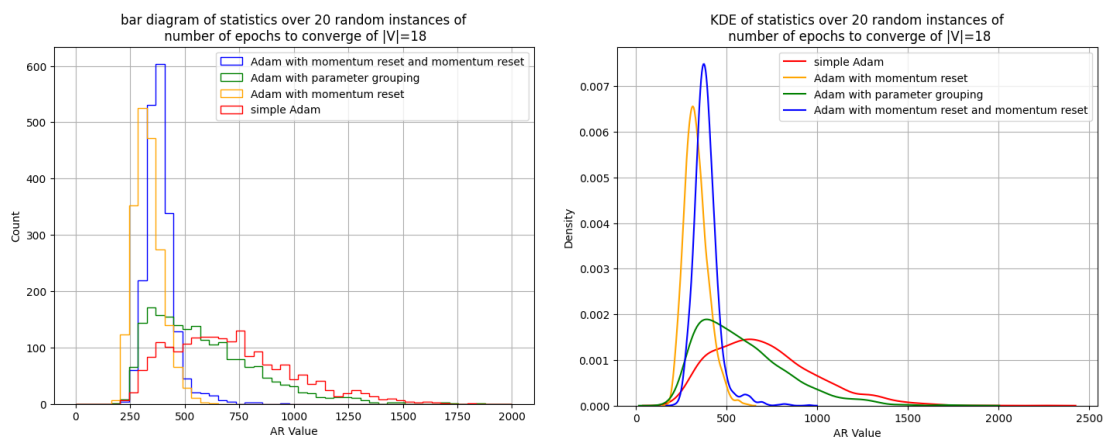


圖 11： $|V| = 18$ 時，20 個隨機 BBV 網絡實例，每個實例取 100 次隨機初始參數的收斂步數之分布直方圖與核密度估計 (KDE) 圖

表 3： $|V| = 12$ 、 $|V| = 15$ 及 $|V| = 18$ 時，20 個隨機 BBV 網絡實例，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$	684.627	318.4535	561.966	366.653
$ V = 15$	685.94	324.1865	568.1655	386.0735
$ V = 18$	711.5395	333.169	586.3435	387.35

VII. 附錄

A. 核密度估計 (KDE) 說明

1. 定義與公式

核密度估計是一種用於平滑樣本分佈的統計方法，可以看作是「直方圖的連續版本」。給定 n 個樣本 $\{x_1, x_2, \dots, x_n\}$ ，KDE 函數公式為：

$$f(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right) \quad (17)$$

其中 $K(u)$ 為核函數，即每個樣本資料點在函數的對應位置上之影響。常見選擇是高斯核 $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$ 。 h 為帶寬，決定平滑程度。 h 越大，核

函數越分散且平緩，曲線越平滑連續； h 越小，核函數越集中且陡峭，曲線越尖銳崎嶇。常見選擇為套用 Scott's Rule 決定，其公式為：

$$h = n^{-\frac{1}{5}} \cdot \sigma \quad (18)$$

其中 σ 為 $\{x_1, x_2, \dots, x_n\}$ 的標準差。

2. 特性與誤解澄清

- 機率密度：

KDE 的其中一種解讀方式為，將 n 個樣本當作同一隨機變數的 n 次獨立抽樣，估計該隨機變數的機率密度 (PDF)。因此他符合 PDF 的特性，例如積分為 1。

- 超過定義域仍有含數值：

如上面的公式，KDE 的實作為「在樣本位置放置一個核函數」，因此樣本位置周邊的函數值也會有所升高。因此若在定義域邊界存在樣本，則該樣本的核函數的「尾巴」可能會超出定義域，此非錯誤，其實是 KDE 的特性之一。

3. 相較於直方圖的優勢

- 易於觀察在連續空間的分布：

KDE 不依賴任意的劃分，能更清晰呈現資料在連續空間上的分布疏密。

- 便於比較：

在實驗資料中，有些在特定區段中的略微優勢可能因為直方圖的劃分被打散，但在 KDE 中則可清楚地看到「有優勢的線條比較高」。

VIII. 參考文獻

- [1] A. Barrat, M. Barthélemy, and A. Vespignani, “Weighted evolving networks: coupling topology and weight dynamics,” *Phys. Rev. Lett.*, vol. 92, no. 22, p. 228701, Jun. 2004.
- [2] J. Choi and J. Kim, “A Tutorial on Quantum Approximate Optimization Algorithm (QAOA): Fundamentals and Applications,” in *Proc. 10th Int. Conf. Information and Communication Technology Convergence (ICTC)*, Jeju Island, Korea, Oct. 2019.
- [3] R. Herrman, P. C. Lotshaw, J. Ostrowski, T. S. Humble, and G. Siopsis, “Multi-angle quantum approximate optimization algorithm,” *Sci. Rep.*, vol. 12, no. 1, p. 6781, Apr. 2022.
- [4] C. Lee, “Adjoint Differentiation,” *PennyLane Demos*, Nov. 2021.
- [5] T. Huang, Z. Zhu, and G. Jin, “Spike-Aware Adam with Momentum Reset for Stable LLM Training,” *arXiv preprint arXiv:2501.06842*, 2025.
- [6] A. Wilkie, J. Ostrowski, and R. Herrman, “An angle rounding parameter initialization technique for ma-QAOA,” *arXiv preprint arXiv:2404.10743*, 2024.