DOI: 10.53104/xdkxts.2025.01.02.005

國產 AI 晶片上的圖優化編譯技術研究——以昇騰 CANN 為例

周映竹 [1]

1. 大連理工大學，遼寧 大連，116024

摘要：國產 AI 晶片的規模越來越大，模型結構也越來越複雜。編譯系統在晶片性能和能效優化中起著重要作用。國產 AI 晶片在硬體設計上已經有了明顯的進步，但性能提升仍受到編譯優化的限制。本文以昇騰 CANN 編譯體系為例，研究圖優化編譯在能效提升和軟硬體配合中的原理和方法。

研究從計算圖結構和編譯理論出發，分析圖優化在運算元融合、圖結構調整和記憶體管理中的作用。圖優化能在演算法和硬體之間建立一種對應關係，使計算更高效。通過分析昇騰 CANN 的體系結構，可以看到它的優化重點是以圖為核心的資源配置策略。它通過運算元融合和資料流程調整，讓執行速度更快，能耗更低。

研究構建了一個多目標約束的能效優化模型，把計算代價、頻寬使用和功率消耗放在同一個分析框架中。這樣可以讓圖優化過程更有規律，也能更好地平衡性能和能耗。

在比較研究中，本文對 CANN、TensorRT、TVM 和 NeuWare 等系統進行了分析。結果顯示，國產編譯系統更注重硬體特徵和能效優化，而國外框架更重視通用性和跨平臺相容性。這種不同說明國產編譯系統處於以性能為主要目標的階段，並在逐步提升開放性。

研究還提出了未來圖優化的發展方向，包括自動優化機制、跨晶片統一編譯框架，以及以能效為重點的自我調整編譯系統。圖優化編譯是提升 AI 晶片算力利用率的重要環節，也是推動國產 AI 算力自主化的關鍵技術。通過更完善和智慧的編譯系統建設，可以在能效、性能和生態相容性之間實現新的平衡。

關鍵字：昇騰 CANN；圖優化編譯；能效建模；運算元融合；NPU 架構；國產 AI 晶片

Research on Graph Optimization Compilation Techniques for Domestic AI Chips: A Case Study of Ascend CANN

ZHOU Ying-zhu[1]

1. Dalian University of Technology, Dalian 116024, P.R.China

Correspondence to: ZHOU Ying-zhu; Email: zzyzhu0112@163.com

Abstract: The scale of domestic AI chips is growing rapidly, and their model structures are becoming increasingly complex. The compiler system plays an important role in improving chip performance and energy efficiency. Although domestic AI chips have made significant progress in hardware design, their performance improvement is still limited by compiler optimization. This paper takes the Ascend CANN compiler system as an example to study the principles and methods of graph optimization compilation in energy efficiency

收稿日期：2025-12-08 **返修日期：**2025-12-23 **錄用日期：**2026-01-08 **出版日期：**2026-01-14
通信作者：zzyzhu0112@163.com
引用格式：周映竹. 國產 AI 晶片上的圖優化編譯技術研究——以昇騰 CANN 為例[J]. 現代科學探索, 2025, 1(2): 49-61.

improvement and software-hardware coordination.

The research starts from the structure of computational graphs and compilation theory, analyzing the role of graph optimization in operator fusion, graph structure adjustment, and memory management. Graph optimization can establish a correspondence between algorithms and hardware to make computation more efficient. By analyzing the architecture of Ascend CANN, it is found that its optimization focuses on a graph-centered resource allocation strategy. Through operator fusion and data flow adjustment, it achieves higher execution speed and lower energy consumption.

A multi-objective constrained energy efficiency optimization model is constructed in this study, integrating computational cost, bandwidth usage, and power consumption into a unified analytical framework. This allows the graph optimization process to be more structured and better balance performance and energy efficiency.

In the comparative study, this paper analyzes systems such as CANN, TensorRT, TVM, and NeuWare. The results show that domestic compiler systems focus more on hardware characteristics and energy efficiency, while international frameworks emphasize generality and cross-platform compatibility. This difference indicates that domestic compiler systems are currently in a performance-oriented stage and are gradually becoming more open.

The study also proposes future directions for graph optimization, including automated optimization mechanisms, unified cross-chip compilation frameworks, and adaptive compilation systems centered on energy efficiency. Graph optimization compilation is a key step to improving the computing power utilization of AI chips and an essential technology for achieving autonomy in domestic AI computing power. Building a more advanced and intelligent compiler system can help establish a new balance among energy efficiency, performance, and ecosystem compatibility.

Key words: Ascend CANN; graph optimization compilation; energy efficiency modeling; operator fusion; NPU architecture; domestic AI chips

引言

人工智慧模型的規模和複雜度不斷增加，對計算能力的需求增長速度已經超過了硬體發展的速度。模型的參數從百萬、億級擴展到千億級，推理和訓練任務對能效提出了更高要求。AI 晶片的性能不再只取決於硬體算力，還取決於編譯系統對計算資源的管理能力。編譯系統連接演算法和硬體，它的優化水準決定了晶片算力的使用效率和整體能耗。國產 AI 晶片在架構設計上進步明顯，但編譯優化的成熟度還需要提升。

已有的 AI 編譯框架，如 NVIDIA 的 TensorRT、Google 的 XLA 和 TVM，已經在 GPU 和異構計算平臺上形成了完整的優化體系。這些框架通常依賴特定的硬體假設，其優化方法與國產 AI 晶片的架構存在差異。國產晶片常使用張量加速單元和多級緩存結構，以支援高密度矩陣計算和高頻寬資料傳輸。這種結構與國外晶片不同，使得傳統編譯框架在移植時無法充分發揮硬體潛能。在圖優化階段，運算元融合、記憶體管理和執行順序優化難以匹配國產晶片的執行方式。結果是硬體算力較強，但實際能效不理想。

為了解決這個問題，國產 AI 晶片廠商開始建立適合自身架構的編譯體系。華為昇騰處理器的 CANN（Compute Architecture for Neural Networks）是一個代表性系統。它採用圖優化驅動的編譯方法，通過運算元融合、圖重構和記憶體複用減少訪存開銷和調度延遲，從而提高 NPU（Neural Processing Unit）的資源利用率。CANN 的出現標誌著國產 AI 編譯技術從指令級優化轉向圖級協同優化的轉變。目前關於 CANN 圖優化機制的研究仍然較少，公開資料多集中在框架使用和運算元開發上，對其優化原理和

能效機制的分析還不充分。

本文以昇騰 CANN 為研究物件，從編譯結構和優化邏輯兩個角度進行分析，探討圖優化在國產 AI 晶片能效提升中的作用和原理。研究目標包括：分析 CANN 圖優化的理論基礎和執行機制，說明其在計算圖層實現軟硬體配合；建立能效建模框架，從計算圖角度研究性能和能耗的關係；對比國際主流編譯系統，分析國產 AI 編譯框架的結構特點和潛在優勢。

本文採用結構分析、演算法建模和比較研究的方法。結構分析揭示 CANN 編譯流程和圖優化的運行邏輯；演算法建模用於表達圖優化的數學模型和能效約束；比較研究以 TensorRT、TVM 等系統為參考，從優化細微性和硬體匹配角度分析 CANN 的特點。本文關注的重點不是框架性能，而是從編譯理論和系統結構的角度，研究國產 AI 編譯系統的能效機制，即通過圖優化提升硬體性能的方式。

1 圖優化編譯的理論基礎

人工智慧模型的計算通常用計算圖（Computation Graph）表示。計算圖通過節點和邊描述運算元之間的資料關係，體現模型的結構，也為硬體執行提供支援。計算圖是一個有向無環圖 G=(V,E)，其中節點 V 表示計算運算元，邊 E 表示張量的資料流程。每個節點對應一個函數 f_i:R^(n_i)→R^(m_i)，邊的連接定義了輸入和輸出的路徑。計算圖的拓撲順序決定執行的依賴關係，所有優化都必須保持運算元功能不變。

在深度學習系統中，計算圖既是資料流程的表達形式，也是編譯系統的重要中間表示（IR）。與傳統編譯器的語法樹不同，計算圖直接反映計算依賴和執行結構，更適合表示大規模並行運算。圖的結構決定模型在硬體上的映射方式。節點數量、邊的密度和圖的深度都會影響運算元調度和記憶體分配。計算圖是模型到機器指令的過渡形式，其結構特徵影響算力的利用效率。

圖優化（Graph Optimization）是在不改變模型功能的前提下，通過結構調整和執行順序改變減少計算量、提升能效的過程。優化可表示為結構映射 G→G'，其中 G'與 G 語義一致，但執行代價函數 C(G')＜C(G)。代價函數包含計算量、記憶體訪問次數和通信延遲等指標，用來衡量性能消耗。圖優化的核心是重構計算路徑，通過依賴分析去除冗餘計算，讓計算方式更符合硬體資源分佈。

在 AI 編譯系統中，常見的圖優化方法包括運算元融合、圖結構重排和記憶體管理三種。

運算元融合（Operator Fusion）把多個相鄰運算元合併為一個執行單元，減少中間張量生成和記憶體訪問。例如，將卷積、歸一化和啟動函數結合，可以在一次計算中完成多個操作，減少內核調用次數和調度成本。

圖結構重排（Graph Reordering）通過調整節點的執行順序或改變依賴路徑來提升並行度或改善資料局部性，讓硬體執行更高效。

記憶體優化關注張量的存儲使用和緩存調度。編譯器可以根據資料依賴關係複用記憶體空間，或提前載入關鍵資料，降低訪存延遲。

這些優化依賴計算圖的可分析性和硬體結構的可建模性。CPU 和 GPU 的圖優化多集中在控制流展開和寄存器使用，而 NPU 的圖優化更偏向資料流程分析。國產 AI 晶片普遍採用張量加速單元、多級緩存和高頻寬互聯結構，圖優化的重點從通用運算元調度轉向片上任務分配。GPU 優化追求通用性，NPU 優化更關注硬體適配和能效控制。

從編譯理論角度看，圖優化是深度學習編譯系統的核心部分。它不是單一模組，而是貫穿圖解析、運算元生成和調度的核心邏輯。圖優化的作用是把計算圖轉化為硬體可執行的指令圖，讓模型結構可以在晶片上高效運行。通過圖優化，編譯系統能在演算法層和硬體層之間建立穩定映射，為能效和資源配置提供基礎[1]。

2 國產 AI 晶片與昇騰 CANN 編譯體系概述

國產 AI 晶片的進步與人工智慧應用的需求擴張幾乎同時出現。自 2017 年起，深度學習

在圖像識別、自然語言處理和大模型訓練等領域得到廣泛應用[2]。國產晶片廠商逐漸形成以計算架構創新和生態建設為核心的發展方向。昇騰、寒武紀、天數智芯、比特大陸等企業在指令設計、算力調度和能效優化上具有共同特徵。第一，晶片普遍使用張量計算單元（Tensor Engine）作為基礎模組，以矩陣乘加為核心運算，提高深度學習中卷積和全連接操作的計算速度。

第二，採用資料流程驅動的調度結構，通過片上緩存和高頻寬互聯減少訪存延遲，提高流水執行效率。

第三，引入分層能效設計，在計算、存儲和通信三部分分配功耗，使晶片在高負載下保持穩定能效。

表 1 展示了當前主要國產 AI 晶片的架構特徵和編譯系統優化方向。

表 1

晶片平臺	製造商	核心架構	峰值算力（TFLOPS）	編譯系統	優化特徵
昇騰 910	華為	Da Vinci NPU 架構	320	CANN	圖優化驅動、運算元融合、能效調度
寒武紀 MLU370	寒武紀科技	TPU-like NPU	256	NeuWare	指令融合、範本運算元調度
比特大陸 BM1684	比特大陸	ASIC 專用架構	196	自研框架	靜態流水線、定制運算元優化
天數智芯 BPU	天數智芯	BPU 異構單元	120	自研框架	頻寬控制與運算元細微性分配

資料來源：各廠商技術白皮書與公開資料。

表中資料表明，國產 AI 晶片多採用 NPU 結構，重點在於提升資料流程效率和能效管理。編譯系統的優化水準決定晶片計算潛能的發揮。與 GPU 的通用編譯方式不同，NPU 依靠編譯器深入分析硬體特性，實現任務映射和圖級優化。運算元融合、圖重排和記憶體調度成為性能提升的主要手段。編譯系統的完善程度已成為評估國產 AI 晶片生態成熟度的重要標準。

華為昇騰系列晶片的 CANN（Compute Architecture for Neural Networks）體系代表了國產 AI 編譯技術的結構化轉型[3]。CANN 不是單一的編譯器，而是面向 AI 計算全流程的編譯與運行框架。它通過建立演算法、運算元與硬體之間的統一抽象層，實現跨晶片版本的一致優化與執行性能。

CANN 體系由四個核心部分組成：前端 IR 層、圖優化模組、運算元庫（TBE）和後端調度引擎。

前端 IR 層負責模型解析與中間表示轉換。主流框架如 TensorFlow、MindSpore、PyTorch 匯出的計算圖會被轉換為 CANN 內部的統一表示結構。該層重點在於保持語義一致與依賴分析，為後續圖優化提供基礎。

圖優化模組是系統的核心部分。該模組通過掃描和重構計算圖，執行運算元融合、資料流程重排和記憶體複用等優化。優化過程包含模式識別、結構重寫和執行計畫生成三個步驟。CANN 採用多級優化策略，局部模式識別用於運算元融合，全域拓撲分析用於圖級調度。層次化設計讓系統能在不同細微性上平衡性能與能效。

運算元庫 TBE 負責運算元範本的定義與代碼生成。每個範本描述運算元的輸入輸出、資料佈局和執行配置。編譯系統根據硬體特性選

擇最合適的實現方式。TBE 支援運算元自動生成和多版本管理，減少人工調優。

後端調度引擎負責將計算圖映射為硬體指令。它根據硬體結構、算力單元分佈和頻寬限制，生成可直接運行的調度圖。該模組與運行時系統配合，實現編譯時優化與運行時調度的閉環。

CANN 的設計思想是以圖優化驅動硬體適配。傳統編譯框架依賴手動調度或靜態範本，而 CANN 通過圖結構分析決定運算元融合、記憶體分配和執行順序。編譯器具備識別硬體特徵的能力，而不是依賴固定規則。例如，在矩陣運算密集的模型中，圖優化模組根據運算元依賴自動調整順序，提高緩存命中率，減少外部訪存能耗。編譯器在生成調度計畫時會動態平衡計算密度和頻寬佔用，使結果在能效與性能間取得較優比例。

CANN 的出現讓國產 AI 編譯體系更系統化。編譯系統不再只追求速度或相容性，而是通過結構化機制提升能效。它連接模型語義、計算結構與硬體執行，為高能效計算奠定基礎，也為後續的圖優化研究提供參考框架。

3 昇騰 CANN 的圖優化機制分析

3.1 編譯體系與圖優化流程

昇騰 CANN 的編譯體系以計算圖為主要結構單元，通過分層設計實現從模型描述到硬體執行的完整映射。系統遵循前端解析、中間優化和後端調度的邏輯。主要模組包括圖前端解析（Graph Engine）、運算元構建模組（TBE Kernel）和任務調度系統（Runtime Scheduler）。圖優化模組貫穿整個編譯過程，負責模型結構調整和資源映射。

在前端階段，CANN 通過 Graph Engine 完成模型的語義解析和圖構建。來自 MindSpore、TensorFlow 或 PyTorch 等框架的模型首先被轉換為統一的中間表示（IR）。該階段保證語義一致和結構完整。Graph Engine 對模型進行拓撲排序，建立節點依賴，並完成類型推斷和運算元分類。每個運算元節點被定義為一個計算單元，包含計算邏輯、輸入輸出資訊和執行約束。這一結構為圖優化階段提供了基礎。

中間層的圖優化模組是 CANN 的核心。它的目標不是單個運算元的微調，而是從整體計算圖角度分析依賴並進行資源映射。優化分為三個步驟：模式識別、圖重寫和計畫生成。

在模式識別階段，系統檢測計算圖中的典型結構，如卷積塊、殘差單元或矩陣乘加鏈，識別可以融合或重排的運算元組。

在圖重寫階段，編譯器根據識別結果調整計算路徑，把獨立節點融合為複合運算元或調整執行順序，減少多餘計算和資料搬移。

在計畫生成階段，系統依據硬體資源約束，如 NPU 核心分佈、存儲容量和頻寬，生成執行圖。計算圖此時被劃分為執行塊（Execution Block），形成硬體可識別的調度計畫。

運算元構建階段由 TBE 模組完成。TBE 將優化後的運算元定義轉換為計算內核（Kernel）。每個內核包含計算邏輯、資料佈局和類型設置。CANN 在此實現運算元範本化和自動生成，讓運算元能根據張量大小和精度動態調整。圖優化和運算元構建互相影響。圖優化確定運算元融合的範圍與輸入輸出關係，TBE 根據結果生成相應內核。運算元執行性能會反過來影響圖優化策略，形成結構與性能之間的迴圈回饋。

圖優化與後端調度之間的銜接體現了 CANN 的系統設計。後端調度模組將優化後的執行圖映射到硬體單元，包括計算核心、存儲資源和通信鏈路。CANN 使用基於硬體拓撲的調度演算法，將執行塊分配到不同 NPU 核心上。調度過程考慮資料依賴與資源使用情況，保持並行度與記憶體佔用的平衡。昇騰的調度機制依賴靜態圖分析結果，執行順序在編譯階段確定，避免運行時調度帶來的額外能耗。

圖優化在 CANN 體系中的作用主要體現在三個方面。

第一，圖優化是模型語義和硬體結構之間的連接中心。沒有圖級優化，運算元編譯無法實現跨層協調，硬體資源會被分散使用。

第二，圖優化是能效控制的主要環節。通過減少訪存、優化資料傳輸和提升緩存命中率，編譯系統在圖層即可完成能耗控制。

第三，圖優化為後端調度提供結構約束。

優化後的執行圖明確任務邊界和依賴，使調度器在編譯階段生成高效執行計畫。

3.2 運算元融合與圖級重構機制

在昇騰 CANN 編譯體系中，運算元融合（Operator Fusion）是圖優化階段的主要策略。它的作用是減少記憶體訪問和任務調度開銷。深度學習模型中包含大量相鄰的線性運算元和非線性運算元，如果這些運算元分別執行，會導致計算單元頻繁交換資料。每次運算元切換都會引起張量的中間存儲、緩存刷新和 DMA（Direct Memory Access）操作，造成性能損失。CANN 通過分析運算元依賴和資料流程特徵，把連續的運算元合併為一個執行單元，使資料直接在片上緩存中傳遞，避免多次訪存。

運算元融合分為兩種類型：線性運算元融合和複合運算元融合。線性融合針對順序連接、資料形態相同的運算元組合，例如矩陣乘加（MatMul + Add）或啟動鏈（ReLU + Dropout）。這些運算元在圖中有固定的依賴關係，融合後形成一個運算核，減少張量創建和釋放次數。複合運算元融合常用於卷積網路，將卷積、批歸一化和啟動函數（Conv + BN + ReLU）組合在一起。這樣卷積結果能直接進入啟動操作，不需要寫入外部存儲。CANN 檢測到這些結構後會自動識別融合機會，並在圖優化中進行結構修改。

融合策略受到網路結構和資料特性的影響。在卷積神經網路中，融合的目標是降低頻寬使用；在全連接網路中，重點是提高運算元複用率和緩存命中率。CANN 在不同網路類型下使用不同融合策略。在 ResNet 網路中，系統更傾向于融合短路徑卷積塊，同時保持跨層連接的獨立性，以保證梯度計算的準確性。在 Transformer 模型中，融合集中在矩陣乘和加法操作，以減輕注意力層的記憶體壓力。這種策略體現了 CANN 對模型結構的識別能力，可以根據圖結構自動選擇合適的融合方式，而不是使用固定範本。

運算元融合減少了訪存次數，也為圖級重構提供了結構基礎。融合後的運算元成為計算密集型節點，輸入輸出依賴更少，便於重排。CANN 利用這一特性，在全域分析階段重新安排依賴路徑。圖重構的主要目標是優化計算順序和資料流程，使任務在硬體上實現最大並行度和更高緩存利用率。

在圖重構中，編譯器建立依賴矩陣，對計算節點進行拓撲排序。系統根據節點的資料量、訪問頻率和依賴深度，動態調整執行順序。例如，當多個運算元共用輸入資料時，系統會讓它們在同一週期執行，以提升緩存命中率。對於依賴路徑較長的計算，編譯器會設置分區點，將任務拆分以減少等待。

CANN 的圖重構結合了靜態優化和硬體限制。NPU 採用流水化執行結構，計算與訪存有嚴格的時間關係。如果計算順序不合理，會造成流水線停頓。圖重構通過分析運算元的執行時間，重新安排資料傳輸和計算次序，使指令和資料傳輸在時間上保持同步。這樣能在不改變模型功能的前提下提高執行效率。

運算元融合與圖重構構成 CANN 圖優化的核心框架。融合減少節點數量和中間資料，圖重構優化執行順序和平衡資源使用。兩種機制共同作用，使計算圖結構更加緊湊，編譯器能夠生成能耗更低、性能更穩定的調度方案。

3.3 記憶體複用與資料流程調度策略

在昇騰 CANN 編譯體系中，記憶體優化和資料流程調度是圖優化的重要部分。NPU 架構不同於 GPU，它的運算元調度更密集，片上存儲限制更嚴格。CANN 在編譯階段通過分析張量的生命週期，結合靜態記憶體規劃和動態複用，建立面向能效的記憶體管理模型。

張量生命週期管理是記憶體複用的基礎。每個計算節點對應多個輸入和輸出張量，它們的存在時間由依賴路徑決定。傳統編譯框架通常採用獨立的分配和釋放策略，每個運算元在執行時單獨申請記憶體，結束後釋放。這種方式實現簡單，但在大型模型中容易產生記憶體碎片並浪費頻寬。CANN 在圖優化階段建立生命週期分析圖（Lifetime Graph），通過遍歷計算圖確定每個張量的起始和結束時刻。系統根據這些時間段計算重疊關係，並把不會同時存在的張量映射到同一物理空間。這樣，張量在執行過程中可以迴圈使用片上存儲，減少記憶體佔用和片外訪問。

靜態配置的優點是執行穩定。編譯階段完成的記憶體映射表能保證執行時不需要額外位址調整。但靜態方法無法處理動態形狀或條件執行結構。CANN 在運行時引入動態複用機制（Dynamic Memory Reuse），由運行時記憶體管理器監控張量狀態。當某個張量計算結束且不再被引用時，系統立即回收空間並分配給新的張量物件。這個過程自動完成，不需要人工干預。靜態規劃提供全域框架，動態複用提供靈活調整，兩種方式組合形成分層記憶體優化體系。

在資料流程調度中，CANN 使用拓撲驅動的調度方法。編譯器通過依賴關係確定執行順序，並在生成執行計畫時進行資料流程重排（Dataflow Reordering）。目標是讓資料在片上盡可能短路徑傳輸，減少 DMA 傳輸次數。系統根據運算元之間的張量複用關係和訪問頻率，對資料流程重新佈線，讓頻繁交互的運算元鄰近執行，減少資料在不同存儲層間的移動。例如，在卷積、批歸一化和啟動函數連續的結構中，CANN 會將三者組合為連續計算區間，使張量只在片上 SRAM 中流動，不進入外部記憶體。

CANN 的資料流程調度依賴硬體拓撲資訊實現軟硬體配合。昇騰 NPU 的內部存儲包括多級緩存和統一控制器，各核心共用片上存儲但頻寬有限。編譯器根據硬體描述檔（Hardware Descriptor）建立頻寬模型，計算各階段的訪存負載。資料流程重排演算法以頻寬均衡為目標，在任務間插入延時指令或調整執行順序，避免訪問衝突。這種方法雖然複雜，但能顯著降低能耗和延遲。

CANN 與 GPU 編譯體系存在明顯差別。GPU 使用統一顯存和多級緩存，優化重點是執行緒並行和訪存合併；NPU 採用固定運算元流水線和任務塊調度，記憶體分配必須在編譯階段確定。GPU 傾向於運行時調優，CANN 強調編譯時確定。GPU 通過動態調度平衡負載，CANN 通過靜態圖優化控制能耗和執行時序。兩種架構的核心區別在執行模型：GPU 依靠多執行緒隱藏延遲，NPU 通過資料流程控制能效。CANN 通過生命週期管理、靜態規劃和動態複用結合，實現了 NPU 資源的精確調度。

3.4 圖優化與硬體協同邏輯

昇騰 CANN 的圖優化機制不僅進行編譯層的結構調整，還直接參與硬體執行的資源配置。其目標是讓計算圖與 NPU 架構實現高效映射，使運算元執行與硬體調度緊密配合。編譯器在這一機制中不只是代碼生成工具，而是控制能耗和性能的核心調度系統。

CANN 在運算元調度階段採用分層映射方法。優化後的計算圖被拆分為多個子圖（Subgraph），每個子圖對應一組關聯度高的運算元。編譯器將這些子圖分配到不同的 NPU 計算單元中，包括 Cube 單元、Vector 單元和 Scalar 單元。運算元類型、資料形式和並行特性決定計算資源配置。例如，卷積運算元主要在 Cube 陣列上執行，資料預處理和啟動函數由 Vector 單元完成。通過這種分層方式，系統在物理結構上實現平行計算。

在運算元調度和指令發射之間，CANN 使用指令依賴分析（Instruction Dependency Analysis）。編譯器依據計算圖拓撲生成依賴矩陣，標識運算元間的順序與資料傳遞。調度模組據此確定指令執行次序，使發射順序與計算圖一致。系統通過硬體指令流（Instruction Stream）執行，以固定步長逐條發射指令，保證不同計算單元同步運行。這種靜態調度方式消除了運行時動態調度的負擔，使能耗分佈更穩定。

片上緩存結構是圖優化與硬體協同的核心。昇騰 NPU 採用多級緩存架構，包括統一緩存（UB）、全域記憶體（GM）和寄存器組。CANN 通過圖優化確定 Tensor 在各層緩存中的駐留位置，使資料流程動和計算過程保持一致。系統在執行計畫生成時，會根據運算元融合與圖重排結果，計算張量訪問頻率和生命週期。高複用數據放在 UB 中，低複用或臨時數據放在 GM 中。編譯器根據緩存大小和並行度調整資料塊劃分，減少計算單元間的訪存衝突。這種規劃方式讓緩存命中率更高，能耗更低。

Tensor 的搬移在 NPU 執行中消耗較多能量。CANN 通過資料流程追蹤機制識別跨運算元或跨子圖的資料路徑，並利用指令融合與資料預取減少搬移成本。當系統檢測到運算元間存在頻繁資料交互時，會生成複合訪存指令，使多個運算元共用一次資料訪問。同時，CANN

預測下一步任務的資料需求，提前觸發 DMA，將資料從 GM 載入到 UB 中，減少等待時間。Tensor 搬移由此從被動操作變為主動調度，使計算與資料流程保持同步。

CANN 的核心思想是讓“計算圖即硬體調度圖”。圖優化不僅調整運算元依賴關係，還定義硬體執行的時間和空間佈局。每次圖結構變化都會影響指令順序、緩存佈局和頻寬使用。編譯器與硬體之間形成動態平衡關係。能效表現正是這種平衡的結果：當圖優化實現高密度資源佔用與低冗余訪存時，晶片功耗自然下降。

與 GPU 相比，CANN 的協同邏輯更具確定性。GPU 依賴運行時執行緒調度和記憶體合併，由硬體自動分配資源。CANN 在編譯階段完成任務劃分和資料路徑規劃。GPU 強調通用性，CANN 強調結構控制與能效優化。GPU 通過大規模執行緒換取性能，NPU 通過結構優化提升能效。CANN 將演算法邏輯嵌入硬體調度過程，使能效控制成為編譯系統的一部分。

4 圖優化的模型化與能效邏輯

4.1 計算圖優化的形式化描述

在編譯系統中，計算圖（Computation Graph）用於表示深度學習模型的結構和依賴關係。它由節點（Node）和邊（Edge）組成。節點表示運算元或張量操作，邊表示資料依賴或控制關係。對 CANN 來說，圖優化的主要任務是在保持模型語義不變的前提下，對圖進行結構調整，使執行代價在硬體限制下最小化。

設計算圖 $G=(V,E)$，其中 $V=\{v_1,v_2,\cdots,v_n\}$ 為運算元集合，$E\subseteq V\times V$ 表示資料依賴集合。每個節點 v_i 具有屬性 $\phi(v_i)=\{t_i,c_i,m_i\}$，分別表示運算元類型、計算代價和記憶體佔用。邊 $e_{ij}=(v_i,v_j)$ 具有權重 w_{ij}，表示資料傳輸的開銷或頻寬消耗。

圖優化問題可以表示為：

$$(\min)\ C(G')=\alpha\sum(v_i\in V')c_i+\beta\sum(e_{ij}\in E')w_{ij}$$

$$\text{"s.t."}\quad \Psi(G')\equiv\Psi(G)$$

其中 G' 為優化後的計算圖，$\Psi(G)$ 表示模型語義結構。約束條件保證優化後圖與原始圖功能一致。目標函數中 α 和 β 為權重係數，用於平衡計算與訪存代價。

這一形式表明，CANN 的圖優化是一個受語義約束的多目標優化問題。變數包括運算元順序、融合方式、張量佈局和資料流程方向。

CANN 的優化可分為三個部分：

（1）圖結構變換（Graph Transformation）：通過模式匹配識別可融合運算元集合 $\Omega=\{S_k\subseteq V\mid f(S_k)\to v_k'\}$，並生成新的節點 v_k'。

（2）記憶體複用規劃（Memory Reuse Planning）：為每個張量 T_i 分配物理空間 $p(T_i)$，當且僅當生命週期區間 $l(T_i)$ 與 $l(T_j)$ 不重疊時，滿足 $p(T_i)=p(T_j)$。

（3）執行順序優化（Execution Ordering）：在保持依賴關係的前提下，尋找拓撲排序 $\pi(V)$，使總代價 $C(G')$ 最小。

這種建模揭示了圖優化的層次結構。上層進行語義等價變換，保證邏輯正確；下層負責資源映射與執行調度，實現物理優化。CANN 在編譯階段通過中間展示層（IR）保持這種分離，讓邏輯優化與硬體約束可以獨立求解。

在具體實現中，CANN 的優化器並不直接求解完整目標函數，而採用啟發式搜索和代價傳播（Cost Propagation）結合的策略。系統先通過圖模式匹配確定候選融合方案，再根據硬體描述檔計算頻寬、緩存和算力參數，估計每種融合的收益，最後選擇代價最低的路徑。這個過程可視為一種動態規劃模型[4]：

$$F(v_j)=(\min)\_\{v_i\in pred(v_j)\}\ \{F(v_i)+\Delta C(v_i,v_j)\}$$

其中 $\Delta C(v_i,v_j)$ 表示節點 v_i 到 v_j 的綜合代價，包括計算、訪存和調度延遲。系統通過代價傳播機制，在全域範圍構建最優執行路徑。

形式化建模讓 CANN 的圖優化具備可量化的分析基礎。計算圖不只是模型轉換的結構，而是演算法與硬體交互的數學描述。編譯器的工作可以看作在語義約束下求解能耗最小化問題。這一思路將圖優化從經驗化的規則體系轉化為可計算的優化過程。

4.2 能效優化的約束模型

在硬體系統中，計算圖的優化必須受資源限制約束。NPU 的性能受到算力峰值、緩存容量和頻寬限制的影響。圖優化中的能效問題可以理解為多重約束下的代價最小化過程，目標是在能耗預算範圍內獲得最優計算效率。

設昇騰 NPU 的資源集合為 R={C_max, B_max, M_max}。其中 C_max 表示計算單元峰值算力，B_max 表示頻寬上限，M_max 表示可用緩存容量。對於優化後的計算圖 $G'=(V',E')$，在編譯階段定義三個代價函數：

$$C\_comp(G')=\sum(v\_i\in V')\, c\_i / C\_max$$

$$C\_mem(G')=\sum(e\_ij\in E')\, w\_ij / B\_max$$

$$C\_stor(G')=(\sum(v\_i\in V')\, m\_i) / M\_max$$

這三個函數分別表示計算利用率、頻寬佔用率和緩存佔用率的歸一化形式。它們構成能效優化的主要約束條件。若以總能量消耗 $E(G')$ 為目標變數，可寫為：

$$(min)\ E(G')=\int_0^T (P\_comp(t)+P\_mem(t))\, dt$$

$$"s.t."\ C\_comp(G')\leqslant 1,\ C\_mem(G')\leqslant 1,\ C\_stor(G')\leqslant 1$$

其中 P_comp(t) 和 P_mem(t) 表示計算與訪存時的功率消耗。該模型說明，CANN 編譯器通過運算元融合和結構調整，使計算與訪存功率保持平衡，而不是追求單一的性能峰值。

在該約束框架中，能效優化可看作多目標規劃問題。定義性能指標 $P(G')$ 與能耗指標 $E(G')$，目標函數為：

$$(max)\ L(G')=\lambda_1 (P(G')/P\_max) - \lambda_2 (E(G')/E\_max)$$

$$"s.t."\ R(G')\leqslant R\_max$$

λ_1 與 λ_2 是權重係數，用於控制性能和能耗的優先順序。CANN 編譯器不會顯式求解該函數，而是通過啟發式代價估計進行間接優化。當運算元融合降低訪存成本時，系統會調整執行批次以控制頻寬；當緩存佔用過高時，系統通過局部圖拆分分配任務，避免功率峰值。該機制是一種基於回饋的多目標優化過程。

能效優化中，關鍵指標是邊際能效收益 $\eta\_i=(\Delta P\_i)/(\Delta E\_i)$。$\eta\_i$ 表示運算元 v_i 的性能提升與能耗變化比值。當 $\eta\_i>1$ 時，優化方案帶來正收益；當 $\eta\_i<1$ 時，系統放棄該優化。CANN 在圖優化階段通過圖級代價函數即時更新 η_i，用於動態判斷優化策略的有效性。通過持續評估 η_i，編譯器能維持性能與能耗的平衡。

CANN 的約束模型與 GPU 系統不同。GPU 依賴運行時功率控制（如 DVFS）調節能耗，而 CANN 在編譯階段即確定部分功率分佈。能效優化不再依靠運行時調節，而是由編譯器結構性決策控制。CANN 在中間展示層（IR）中嵌入能耗權重，使圖優化與能量管理保持統一邏輯。編譯器生成的執行圖決定任務順序，也決定能量流向[5]。

約束模型使圖優化從局部性能調整擴展為系統級能效配置。CANN 的優化過程在多個資源維度中尋找平衡點，這個平衡由硬體條件決定，也受編譯器策略影響。能效因此成為系統設計的一部分，而不是計算結果。

4.3 靜態優化與動態優化的比較邏輯

在 AI 編譯系統中，計算圖的執行方式主要分為靜態圖（Static Graph）和動態圖（Dynamic Graph）。兩種方式的區別在於調度時機和能效管理方法。靜態優化在編譯階段完成所有圖結構優化、運算元融合和記憶體調度，運行時執行路徑固定。動態圖在運行時根據輸入或環境變化動態生成計算圖，可即時調整執行計畫。

在能效管理上，兩種方式表現出明顯差異。靜態圖優化的能耗模型在編譯階段可完全展開。編譯器根據全域拓撲結構計算每個運算元的訪存量、算力需求和張量生命週期，並在編譯階段完成能量分配。靜態優化的優勢包括：

（1）能量預算和算力調度在執行前已確定，硬體在運行初期即可鎖定電壓和頻率，減少波動；

（2）資料流程路徑固定，緩存複用率高，訪存能耗更低；

（3）執行順序穩定，可實現高並行而無額外調度開銷。

這些特徵使靜態優化適合昇騰 NPU。該架構以批次處理和張量並行為核心，依賴確定的調度計畫實現高算力密度。

動態圖的能效表現依賴運行時控制。動態圖框架（如 PyTorch、DyGraph）通過即時構圖提供靈活性，但每次圖生成都帶來額外的調度和資源消耗。運行時的記憶體分配、運算元編譯和依賴解析佔用算力，使功率曲線波動。對於高吞吐 NPU，這種不確定性導致能耗波動增大，功率規劃難以保持穩定。

CANN 採用靜態優化路徑，有明確的技術原因。昇騰架構使用多級緩存和片上匯流排，資源劃分固定。靜態圖能在編譯階段確定運算元順序和資料佈局，使片上資源使用最優。CANN 以運算元範本（TBE）和固定圖結構為基礎，通過離線優化生成執行計畫，提升不同模型間的性能複用。靜態圖的確定性還便於功率控制。編譯器在編譯期預測每個階段的功率分佈，系統預設電源與散熱參數，使能效曲線平穩。

靜態優化的局限在於缺乏輸入自我調整能力。不同批量和輸入形狀下，固定圖結構難以充分利用硬體潛力。它也無法在運行中調整優化策略，一旦圖結構鎖定，系統不能依據負載變化重新調度。靜態優化對模型結構變化敏感，輕微修改都會觸發重新編譯，影響部署效率。這些問題在多工推理和流式資料處理中尤為明顯。

一種新的優化方向是自我調整圖優化（Adaptive Graph Optimization）。這種方法結合靜態和動態機制，在保持靜態優化能效的基礎上引入運行時調整能力。其實現方式分為三個層次：

第一，參數化圖結構。在中間展示層（IR）中定義可調節參數，如張量塊大小和並行度閾值，使編譯器在運行時根據硬體回饋調整執行分支。

第二，能效回饋機制。在執行單元中採集功率和延遲資料，將結果回饋給運行時調度模組，用於修正編譯期的代價模型。

第三，增量式優化編譯。保留部分中間優化狀態，實現快速再編譯，而不是完全重新構圖。這樣可在輸入變化時維持高能效。

這種方法為靜態編譯體系引入了動態回應能力。它不改變 CANN 的基本結構，而是在確定性編譯框架中加入能效調節功能，使系統在保持穩定性的同時具備適應性。未來的圖優化系統可能採用分層結構：底層由靜態圖維持硬體穩定，上層由輕量運行時實現策略更新，在能效與靈活性之間取得平衡。

5 國產 AI 編譯系統的比較與反思

國產 AI 編譯系統的設計方向與國際主流框架不同。昇騰 CANN 的核心特徵是以硬體為中心的圖優化邏輯，通過精確建模硬體結構提升性能和能效。國際系統如 TensorRT 和 TVM 注重通用性與跨平臺能力，追求在不同硬體之間維持一致的抽象介面和優化邏輯。這種差異反映了“硬體專用化”和“軟體可攜性”的技術取向。

表 2 展示了 CANN 與主流 AI 編譯系統的差異。

表 2

編譯框架	開發主體	圖優化細微性	融合策略	IR 層次	硬體依賴性	優化目標
CANN	華為	圖級 + 運算元級	自動融合與模式匹配	接近硬體層	強	性能與能效平衡
TensorRT	NVIDIA	運算元級	範本融合	中層	強	推理時延最小化
TVM	Apache	圖級	Pass 重寫 + AutoTVM	高層	弱	通用性與編譯速度

NeuWare	寒武紀科技	運算元級	固定融合範本	中層	強	硬體利用率最大化

CANN 的優化邏輯貼近硬體語義，耦合度高。編譯器在編譯階段通過靜態規劃確定資料流程路徑與 NPU 指令調度，使硬體資源利用最大化。代價模型由硬體描述檔驅動，反映存儲結構、匯流排拓撲和功率限制。TensorRT 使用啟發式方法，以推理時延最小為目標，主要針對卷積、歸一化、啟動運算元優化。系統在運行時動態選擇 kernel variant 以匹配 GPU 架構。TVM 利用 Relay 中間表示將計算圖分解為可重寫運算式，再結合自動調優實現性能搜索，目標是通用性和編譯速度。NeuWare 採用半靜態圖優化，通過範本運算元描述與運行時重構結合實現性能調節。

CANN 的運算元融合細微性更細，直接受硬體流水線深度和緩存容量影響。編譯器在融合時需滿足資源限制。例如在 Conv-BN-ReLU 鏈中，CANN 根據緩存容量決定是否合併 BN 運算元，以避免訪存衝突。TensorRT 的融合偏向語義匹配，不考慮底層硬體約束。TVM 通過模式匹配實現跨層融合，但受 IR 層級限制。NeuWare 使用預定義範本，編譯效率高但缺乏靈活性。CANN 的策略體現“結構約束下的靈活性”，即在硬體邊界內啟發式選擇局部最優方案。

各系統在中間表示（IR）層次上也存在差異。CANN 的 IR 包含張量維度、訪存拓撲與任務映射資訊，優化器可訪問硬體描述符執行指令級調度。TVM 的 Relay IR 更接近演算法表達層，抽象出高層語義，不直接描述硬體特徵。TensorRT 的 IR 位於中間層，部分結合 GPU 特徵。NeuWare 的 IR 與 CANN 相似，但依賴人工設定檔。

這種差異帶來優化能力和通用性的權衡。IR 越接近硬體，優化精度越高，但跨平臺能力弱。IR 越抽象，通用性強，但難以實現精細調度。CANN 選擇靠近硬體的路徑，是因為當前國產晶片仍處於算力優化階段，優先追求性能和能效。為實現極限能效，犧牲部分通用性是一種工程上的合理選擇。

以硬體為導向的設計也存在問題。CANN 在不同晶片版本間相容性差，硬體升級時需要重新定義運算元範本和優化規則。封閉的生態體系限制外部框架的接入，科研與產業應用之間存在介面缺口。相比之下，TVM 等開源框架通過高層抽象實現快速適配，使新硬體能更快融入生態系統。

未來國產編譯系統可在專用化與通用化之間尋求平衡。一種方向是構建分層編譯架構。上層保留通用 IR 和自動調優機制，下層開放硬體介面以支援針對性優化。開放部分優化介面，讓外部研究者參與演算法優化，可增強生態協作與創新能力。

6 圖優化編譯的趨勢與制度化構想

圖優化編譯技術正在從工程經驗向體系化智慧發展。AI 模型的複雜度提高，硬體架構日益多樣，傳統依賴人工規則和靜態調優的編譯方式難以兼顧算力與能效。未來的編譯系統需要具備自學習、自調節和跨平臺協作能力，使圖優化成為可持續進化的過程，而不是固定規則的堆積。

6.1 自動化圖優化的演化方向

傳統圖優化依靠人工定義運算元融合規則和固定調度範本。性能受限於專家經驗和單一硬體特徵。自動化圖優化成為新的研究重點。未來的圖優化將利用機器學習方法，通過性能資料驅動的回饋機制實現自動調優和自動融合。系統在第一次編譯時採集運算元執行資訊、訪存延遲和能耗資料，建立性能模型。然後，使用強化學習或貝葉斯優化演算法搜索最優融合和調度方案。這種學習式優化方式能在不同晶片和模型間遷移，形成自更新的優化知識庫。

在昇騰 CANN 體系中，這一趨勢已有體現。CANN 部分模組支援基於代價模型的自動融合判斷，但仍依賴人工規則。如果引入基於性能資料的回饋學習機制，編譯器可以在保證語義等價的前提下自動提煉模式，從而在不同 NPU

架構間快速適配。核心挑戰是構建統一的性能度量標準和可泛化的優化表示。只有在性能可對比、優化可複現的條件下，自動化優化才能進入制度化階段。

6.2 跨晶片統一優化框架的構想

國產 AI 晶片生態呈多架構並行格局。昇騰、寒武紀、比特大陸、天數智芯等體系在指令集、記憶體架構和運算元庫上差異明顯，編譯系統各自獨立。模型在不同晶片間遷移時需重新編譯和調優，造成高成本。這種結構性分裂降低了整體效率，也影響生態協同。

建立跨晶片統一優化框架是制度化圖優化的關鍵。核心目標是以中立化中間表示（IR）和抽象硬體描述介面為基礎，實現編譯器間的互通與複用。

框架可分為三層：

1) **統一中間展示層（Unified IR）**：抽象計算圖的邏輯結構與資料依賴，不包含具體硬體特徵。

2) **硬體描述層（Hardware Abstraction Layer）**：通過標準介面描述 NPU 資源特性，如指令延遲、緩存容量和功率曲線。

3) **自我調整優化層（Adaptive Optimizer）**：根據 IR 與硬體描述生成優化策略，執行圖重構、運算元融合與能效調度。

這一框架與 CANN 現有閉環優化體系相相容，是國產生態的擴展。若能建立通用介面標準，使不同廠商的編譯器在 IR 和運算元層共用優化邏輯，可顯著提升模型遷移效率，減少重複調優。統一性不是削弱差異，而是通過制度化建立協同基礎，使硬體差異成為優化約束而非障礙。

6.3 面向“能效智慧體”的編譯系統結構

傳統編譯器的任務是將演算法轉化為可執行代碼。隨著 AI 計算能耗成為核心問題，編譯器正在從“性能優化工具”轉變為“能效管理核心”。未來的圖優化編譯系統可形成三層結構：

1) **軟硬體協同層**：編譯器與硬體運行時系統建立回饋通道。晶片即時報告功率、溫度和延遲，編譯器據此調整運算元調度和資料路徑，實現能效閉環控制。

2) **能效約束層**：在圖優化階段引入能量預算，將能耗、延遲和算力作為等權目標，使用多目標演算法生成平衡方案，使能耗成為編譯決策參數。

3) **自我調整優化層**：結合歷史資料和任務特徵，系統識別能效模式，在後續執行中優先選擇最優配置。編譯器因此具備學習與記憶能力，形成持續優化機制。

這種結構讓編譯系統從靜態優化器變為能效智慧體。它成為軟硬體協調的主動核心。對於昇騰 CANN，這意味著編譯系統不再是工具鏈元件，而是晶片能效控制中心。通過與功率管理單元（PMU）和監控模組連接，編譯器可在任務級別調整運算元並行度和功率分配，實現“軟體控制硬體”的動態優化。

7 結論

本研究以昇騰 CANN 為研究物件，對圖優化編譯的理論邏輯和能效機制進行了系統分析。通過對計算圖結構、運算元融合、記憶體調度和硬體協同的分析，揭示了 CANN 以圖優化為核心驅動力的設計思路。其優化邏輯體現出從演算法語義到硬體執行的映射關係，通過靜態圖重構和資源約束分析，實現軟硬體間的高效配合。這種機制的重點不在於單一演算法性能，而在於利用編譯系統把能效約束納入 AI 計算的結構框架。

在理論部分，本研究提出了計算圖優化與能效建模的分析框架。該框架基於圖論和多目標優化理論，將執行代價、頻寬使用和能耗指標放入同一模型，用來解釋 CANN 在運算元級與系統級的能效調度邏輯。這種框架不同于傳統的性能優先編譯方式，更關注約束下的平衡，使能效優化變成可建模、可分析的過程。這一方法為國產 AI 晶片編譯系統的優化研究提供了理論支援。

在系統比較中，通過分析 TensorRT、TVM 和 NeuWare 等框架，可以看到國產編譯系統的主要取向。以硬體特徵為核心的定制化優化在性能上有明顯優勢，但通用性和生態相容性較

弱。這種現象不是技術缺陷，而是產業階段的特徵。未來的方向是尋求性能與開放性的平衡，使編譯系統既能體現硬體特徵，又能支援多晶片協同與生態統一。

研究仍存在局限。昇騰 CANN 的部分機制未公開，特別是調度與功率約束模型的細節，公開資料有限，理論驗證仍有不確定性。能效模型的驗證主要依賴模擬環境，實際資料的系統分析需要進一步補充。

未來的圖優化編譯將進入能效自調節與智慧優化階段。編譯器不再只是性能工具，而是能效智慧體，負責算力調度、能量分配和模型自我調整。對中國而言，國產 AI 編譯系統的完善不僅是技術問題，也是算力自主的關鍵路徑。是否能在圖優化層面建立制度化、開放化和智慧化體系，將直接影響 AI 計算生態的競爭力。

參考文獻：

[1] Chen T, Moreau T, Jiang Z, Zheng L, Yan E. TVM: An Automated End-to-End Optimizing Compiler for Deep Learning[C]// Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’ 18). USENIX Association, 2018: 578-594.

[2] 張芳芳. 基於 CNN 加速器的深度學習編譯器設計與實現[D]. 西安: 西安電子科技大學, 2020.

[3] 《華為研究》編輯部. 科學與工業中的 AI 應用及其前景[J]. 華為研究·計算專刊, 2024(6): 12-25. 深圳: 華為技術有限公司.

[4] Ding Y, Yu C H, Zheng B, Liu Y, Wang Y, Pekhimenko G. Hidet: Task-Mapping Programming Paradigm for Deep Learning Tensor Programs[EB/OL]. arXiv:2210.09603, 2022.

[5] Furutanpey A, Walser C, Raith P, Frangoudis P A, Dustdar S. Leveraging Neural Graph Compilers in Machine Learning Research for Edge – Cloud Systems[EB/OL]. arXiv:2504.20198, 2025.

版權聲明

出版聲明