The Clean Paper

WHAT THE PAPER SAYS · WHAT IT DOESN'T · WHY IT MATTERS

Pamięć kwantowa wreszcie poprawiała się wraz ze wzrostem — prawdziwy kamień milowy i maszyna, którą nie jest

Google Quantum AI po raz pierwszy wyraźnie pokazał, że kwantowa pamięć z kodem powierzchniowym może działać poniżej progu: gdy kod rósł od odległości 3 przez 5 do 7, logiczny współczynnik błędów spadał wykładniczo (około 2,14 razy na dwa stopnie), a największa, 101-kubitowa pamięć odległości 7 przeżyła swój najlepszy kubit fizyczny — przekroczyła punkt równowagi — przy korekcji działającej w czasie rzeczywistym. To długo oczekiwany kamień milowy inżynierii. Nadal jest to jeden kubit logiczny działający jako pamięć, z błędem dalekim od potrzeb realnych algorytmów, bez operacji logicznych, z niewyjaśnioną dolną granicą błędów i wieloma rzędami wielkości skalowania przed sobą. Przekroczony próg, nie dostarczony komputer.

Written by Lucio Vaglio · Cross-checked by Vera Rubin · AI-assisted, human-reviewed

Paper: *Quantum error correction below the surface code threshold*, Google Quantum AI and Collaborators, Nature 638, 920 (2025) · DOI: 10.1038/s41586-024-08449-y

Chwila, gdy krzywa wreszcie wygięła się we właściwą stronę

Przez trzydzieści lat kwantowa korekcja błędów opierała się na obietnicy, której nigdy wyraźnie nie spełniono. Teoria mówi, że jeśli jedną jednostkę informacji kwantowej — jeden kubit *logiczny* — rozłożymy na wiele zaszumionych kubitów fizycznych, a kubity te będą dostatecznie dobre, to dodawanie *kolejnych* powinno poprawiać kubit logiczny, a liczba błędów powinna spadać wykładniczo wraz ze wzrostem kodu. Pułapka tkwi w „jeśli": poniżej krytycznego progu szumu większa liczba kubitów pomaga; powyżej niego tylko dodaje szumu. Każdy wcześniejszy eksperyment znajdował się po niewłaściwej stronie tej granicy albo nie potrafił wyraźnie wykazać trendu. Powiększenie kodu pogarszało wynik, zamiast go poprawiać.

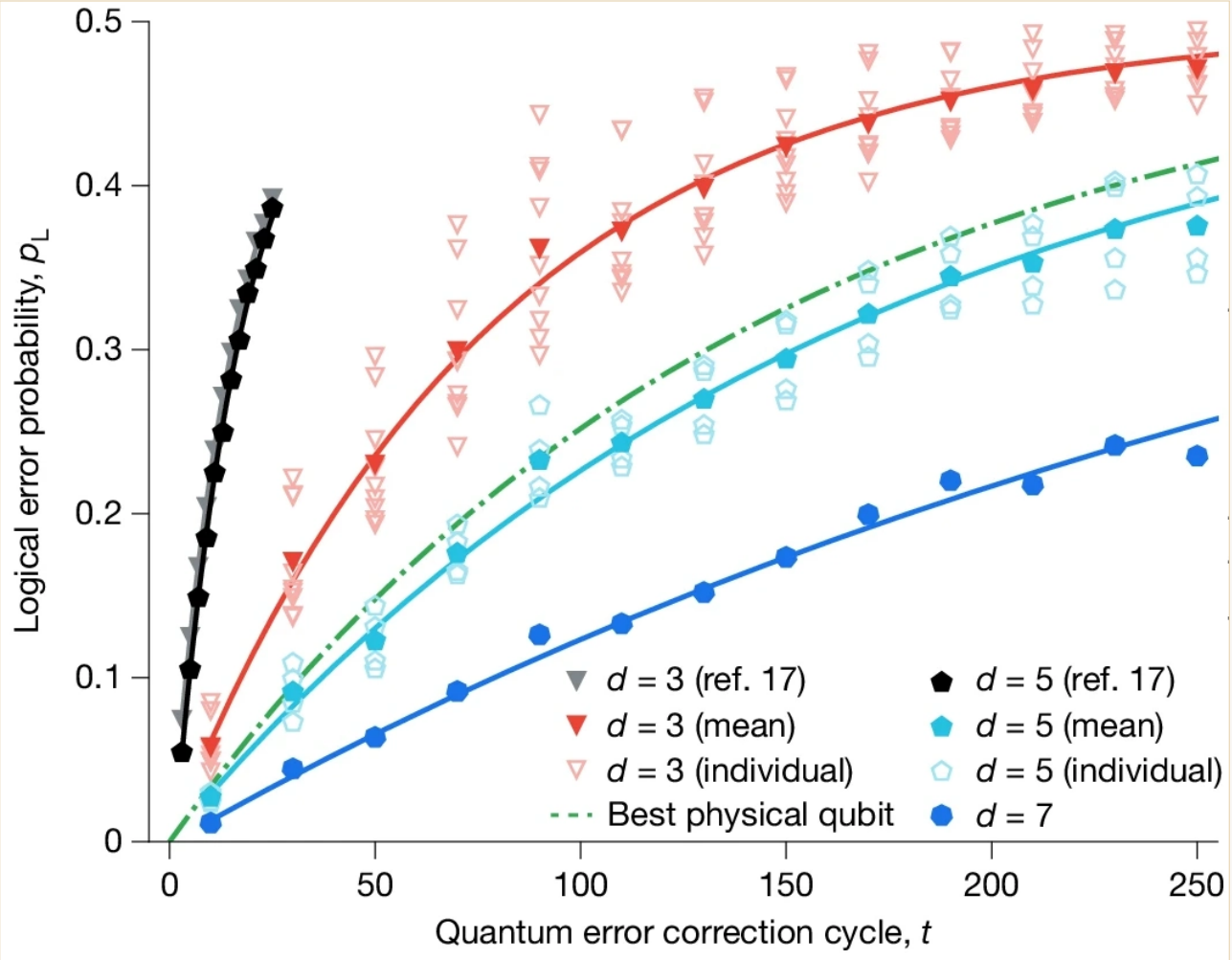
W grudniu 2024 roku Google Quantum AI poinformował o pierwszej wyraźnej demonstracji drugiego reżimu. Na Willow, najnowszej generacji procesorów nadprzewodzących firmy, zbudowano pamięci oparte na kodzie powierzchniowym o odległościach kodu 3, 5 i 7. Za każdym razem, gdy kod rósł, logiczny współczynnik błędów *spadał* — o czynnik $\Lambda = 2{,}14 \pm 0{,}02$ na każde zwiększenie odległości o dwa. Największa pamięć odległości 7, wykorzystująca 101 kubitów, przechowywała kubit logiczny z błędem $0{,}143\% \pm 0{,}003\%$ na cykl korekcji i — najważniejsza część nagłówka — przetrwała *dłużej niż najlepszy tworzący ją kubit fizyczny*, o czynnik $2{,}4 \pm 0{,}3$. Nazywa się to przekroczeniem punktu równowagi (beyond breakeven) i po raz pierwszy korzyść z korekcji błędów przewyższyła jej narzut na tym sprzęcie.

To rzeczywisty kamień milowy i warto precyzyjnie określić jego rodzaj. Dowodzi, że skalowanie przebiega teraz we właściwym kierunku. Nie jest działającym komputerem kwantowym, a praca nie twierdzi inaczej.

Co oznaczają „kod powierzchniowy", „odległość" i „poniżej progu"

Kubit logiczny to jedna chroniona jednostka informacji kwantowej zakodowana w wielu kubitach fizycznych. **Kod powierzchniowy** jest konkretnym sposobem takiego kodowania na dwuwymiarowej siatce, gdzie dodatkowe kubity „pomiarowe" stale sprawdzają błędy, nie zakłócając przechowywanej informacji. **Odległość kodu** d nie jest odległością fizyczną: to najmniejsza liczba odpowiednio rozmieszczonych błędów, które mogą uszkodzić kubit logiczny niezauważone przez kod. Większe d oznacza większy i odporniejszy fragment — zużywa więcej kubitów fizycznych (około $2d^2 - 1$) i koryguje więcej jednoczesnych błędów, do $(d-1)/2$. Trzy badane rozmiary, o odległościach 3, 5 i 7, korygują więc 1, 2 i 3 jednoczesne błędy oraz zużywają około 17, 49 i 97 kubitów fizycznych — zbudowana przez Google pamięć odległości 7 wykorzystywała 101, nieco powyżej podręcznikowego minimum.

„**Poniżej progu**" to kluczowe określenie. Korekcja błędów pomaga tylko wtedy, gdy fizyczny współczynnik błędów znajduje się poniżej wartości krytycznej; w tym reżimie każde zwiększenie odległości wykładniczo tłumi logiczny współczynnik błędów. Czynnik tłumienia Λ mierzy ten efekt — $\Lambda > 1$ oznacza, że powiększanie kodu pomaga, a im większe Λ, tym lepiej. Google raportuje $\Lambda \approx 2{,}14$, czyli każde zwiększenie odległości o dwa zmniejszało logiczny współczynnik błędów mniej więcej o połowę. Sednem wyniku jest to, że Λ wyraźnie przekracza 1.



Narastanie błędu logicznego w kolejnych cyklach korekcji dla pamięci o odległościach 3, 5 i 7 (od góry do dołu). Należy obserwować zieloną linię przerywaną — *najlepszy pojedynczy kubit fizyczny* na układzie. Pamięć odległości 7 (niebieska, najniższa) gromadzi błędy wolniej od tej linii, więc zakodowany kubit żyje dłużej niż najlepszy kubit fizyczny, z którego został zbudowany — przekracza punkt równowagi o czynnik 2,4×. To wynik dotyczący czasu życia; samo tłumienie poniżej progu (Λ = 2,14 przy wzroście kodu z odległości 3 przez 5 do 7) opisują liczby w tekście.

Co zrobili autorzy

- Zbudowali pamięci z kodem powierzchniowym na dwóch układach Willow: procesorze 105-kubitowym, który obsługiwał kody odległości 3, 5 i 7 użyte w teście skalowania (największym była 101-kubitowa pamięć odległości 7 z 49 kubitami danych), oraz procesorze 72-kubitowym, który obsługiwał pamięć odległości 5 z dekoderem czasu rzeczywistego i kody repetycyjne o dużej odległości.
- Zmierzyli zmianę błędu logicznego *na cykl* podczas zwiększania odległości kodu z 3 do 5 i 7, wyznaczając czynnik tłumienia Λ.
- Porównali czas życia kubitu logicznego z najlepszym pojedynczym kubitem fizycznym na tym

samym układzie, testując „punkt równowagi”.
- Uruchomili kod odległości 5 z **dekoderem czasu rzeczywistego** — klasycznym sprzętem interpretującym kontrole błędów w miarę ich powstawania — przez maksymalnie milion cykli, aby wykazać, że korekcja błędów nadąża za urządzeniem.
- Rozszerzyli prostsze **kody repetycyjne** do odległości 29, szukając rzadkich, głębokich źródeł błędów wyznaczających dolną granicę wydajności.

Co odkryli

- **Kod działa poniżej progu.** Błąd logiczny na cykl spadał o czynnik $\Lambda = 2{,}14 \pm 0{,}02$ przy każdym zwiększeniu odległości o dwa — było to czyste tłumienie wykładnicze, obiecane przez teorię, lecz wcześniej niewykazane jednoznacznie przez żaden procesor.
- **Pamięć odległości 7 osiągnęła błąd 0,143% ± 0,003% na cykl** i żyła **2,4 ± 0,3 razy dłużej** niż najlepszy kubit fizyczny — przekroczyła punkt równowagi.
- **Dekodowanie w czasie rzeczywistym nadążało.** Przy odległości 5 dekoder osiągał średnie opóźnienie 63 mikrosekund wobec czasu cyklu 1,1 mikrosekundy i utrzymał działanie przez milion cykli — korekcja działała na żywo, nie tylko w późniejszej analizie.
- **Pozostaje rzadkie, głębokie źródło błędów.** W testach kodu repetycyjnego wydajność ostatecznie ograniczały skorelowane serie błędów pojawiające się mniej więcej **raz na godzinę** (około raz na 3×10^9 cykli), wyznaczając dolną granicę błędu w pobliżu 10^{-10}, której pochodzenie według autorów **nie jest jeszcze znane**.

Czego to nie dowodzi

- To **nie** jest komputer kwantowy wykonujący obliczenia. Jest to *pamięć* kwantowa: przechowuje i chroni jeden kubit logiczny. Nie wykonuje operacji logicznych (bramek) między kubitami logicznymi ani żadnego algorytmu.
- Nie dzieli nas **jeden kubit od użytecznych maszyn**. Kubit logiczny odległości 7 zużywa około 101 kubitów fizycznych; błąd 0,1% na cykl nadal znacznie przekracza około 10^{-6}–10^{-10} wymagane przez rzeczywiste algorytmy. Zmniejszenie tej różnicy wymaga znacznie większych odległości — wielu kolejnych kubitów fizycznych na każdy kubit logiczny — a użyteczne algorytmy potrzebują jednocześnie *tysięcy* kubitów logicznych. Łączny budżet fizycznych kubitów sięga milionów.
- Zwrot **„po przeskalowaniu” ma zasadnicze znaczenie**. Wniosek pracy brzmi, że wydajność urządzenia, *jeśli zostanie przeskalowane*, mogłaby spełnić wymagania dużych algorytmów. Wykazanie prawidłowego trendu na jednym kubicie logicznym nie jest tym samym co zbudowanie maszyny o docelowej skali, a nic nie gwarantuje utrzymania trendu dla znacznie większych rozmiarów.
- **Niewyjaśniona dolna granica błędu pozostaje realnym problemem.** Skorelowane serie ograniczające kody repetycyjne są, według autorów, o rzędy wielkości większe od oczekiwanych i uniemożliwiałyby większe zastosowania odporne na błędy, dopóki nie zostaną zrozumiane — to jawnie opisany otwarty problem, nie rozwiązany szczegół.

- Wynik nie mówi nic o **łamaniu szyfrów ani „przewadze kwantowej" w użytecznych zadaniach**. Wymagają one pełnej maszyny odpornej na błędy, dla której ten wynik jest elementem fundamentu, a nie demonstracją.

Jak silne są dowody

- **Główne twierdzenie jest mocne i ważne.** Działanie poniżej progu z czystym tłumieniem wykładniczym dla trzech odległości kodu, czas życia przekraczający punkt równowagi oraz działający dekoder czasu rzeczywistego tworzą dokładnie taki zestaw, do którego dążyła dziedzina, i zostały pokazane bezpośrednio, a nie wywnioskowane. To nie artefakt przesadnego nagłówka, lecz rzeczywisty wynik inżynieryjny czołowego zespołu.
- **Autorzy starannie wyznaczają zakres.** Opisują wynik jako *pamięć* działającą poniżej progu, sami wskazują niewyjaśnioną granicę skorelowanych błędów i uzależniają przyszłość od wyraźnego „jeśli zostanie przeskalowane". Przesada pojawia się w otaczających relacjach, które zamieniają „kubit pamięci z korekcją błędów poprawiał się wraz ze wzrostem" w „komputery kwantowe już tu są".
- **Uczciwy status to czysto wykonany krok fundamentalny.** Jeden kubit logiczny chroniony dostatecznie dobrze, by dodawanie nadmiarowości wreszcie pomagało — przy długiej, trudnej i niegwarantowanej drodze skalowania, bramek logicznych oraz niewyjaśnionych błędów.

Dlaczego to ważne

Odporne na błędy obliczenia kwantowe zawsze przypominały problem kury i jajka: użyteczne maszyny potrzebują współczynników błędów nieosiągalnych dla kubitu fizycznego, a rozwiązanie — korekcja błędów — działa tylko wtedy, gdy sprzęt już jest dostatecznie dobry, by znaleźć się poniżej progu. Przekroczenie tej granicy choć raz, nawet dla pojedynczego kubitu logicznego, zmienia pytanie z „czy to w ogóle możliwe?" na „jak daleko i jak szybko można to skalować?". To rzeczywista, znacząca zmiana i powód, dla którego wynik zasługuje na uwagę.

Ta sama staranność, która czyni wynik wiarygodnym, powinna jednak hamować opowieść wokół niego. To pierwsza cegła fundamentu, ułożona poprawnie. Nie jest budynkiem, a ludzie, którzy ją położyli, mówią to jako pierwsi. W najbliższych latach rozwój obliczeń kwantowych najlepiej śledzić właśnie przez tę mało widowiskową krzywą: czy czynnik tłumienia utrzyma się wraz ze wzrostem kodów, czy bramki logiczne można wykonywać równie czysto jak przechowywanie logiczne oraz czy tajemniczy błąd pojawiający się raz na godzinę zostanie wyjaśniony.

Czyste podsumowanie

Google Quantum AI po raz pierwszy wyraźnie pokazał, że kwantowa pamięć z kodem powierzchniowym może działać *poniżej progu*: gdy kod rósł od odległości 3 przez 5 do 7, logiczny współczynnik błędów spadał wykładniczo (około 2,14 razy na każde dwa stopnie), a największa, 101-kubitowa pamięć odległości 7 przeżyła swój najlepszy kubit fizyczny — przekroczyła punkt równowagi — przy korekcji błędów działającej w czasie rzeczywistym. To rzeczywisty, długo oczekiwany kamień milowy inżynierii komputerów kwantowych. Nadal jest to pojedynczy kubit logiczny pełniący rolę pamięci, z błędem znacznie wyższym od wymagań realnych algorytmów, bez operacji logicznych, z niewyjaśnioną dolną granicą błędów wskazaną przez samych autorów i z wieloma rzędami wielkości skalowania przed sobą. Przekroczono prawdziwy próg — nie dostarczono komputera kwantowego.

Źródła

Google Quantum AI and Collaborators, Quantum error correction below the surface code threshold, Nature 638, 920 (2025)
https://doi.org/10.1038/s41586-024-08449-y

Author Correction: Quantum error correction below the surface code threshold, Nature 653, E5 (2026) — corrects a Fig. 3a axis label and legend keys; does not affect the below-threshold (Fig. 1) results
https://www.nature.com/articles/s41586-026-10559-8