Repeated six-qutrit layers for historical FX forecasts

A matched depth comparison and a trained-state inspection

PhaseWeave research note | 21 November 2026

Abstract. We vary circuit depth at fixed parameter count in a graph transformer that forecasts next-second foreign-exchange cycle margins in both directions. Four entangling layers reduce mean absolute error relative to one on the reused November afternoon and September stress panels. They increase error on the reused CHF panel. A one-layer coherent product circuit attains the lowest two-seed mean error on every panel. Independent tensor contractions reproduce amplitudes and parameter gradients from the trained entangling states. The comparison measures historical forecasts.

Matched experiment. A width-200 graph transformer takes 16 successive graph frames and produces two one-second forecasts. The six-wire patch maps each frame to six local clock expectations. All 24 models have 1,538,073 trainable parameters including 1,447 in the patch, and use the same initial parameter arrays within each seed. We cross depth $L \in \{1, 2, 4\}$, four propagation arms, and two fixed seeds. Each fit takes 384 updates in three epochs, on 8,192 August training windows. A disjoint 2,048-window September panel selects the checkpoint. Epoch three was selected for 22 fits and epoch two for two. The graph tokenizer follows the node and edge token principle of TokenGT [1].

Repeated state evolution. Let $c(s) = \cos(2\pi s/3)$, let x_i denote the learned angle for wire i, and let $p_i(s)$ be the softmax over the logits $\cos(x_i - 2\pi s/3)$. For level tuple $\mathbf{s} \in \{0, 1, 2\}^6$, the prepared amplitude is $\psi_0(\mathbf{s}) = \prod_i \sqrt{p_i(s_i)}$. With learned bond strengths λ_i, apply

$$D_{\mathbf{s},\mathbf{s}} = \exp\left(\mathrm{i} \sum_{i=1}^{6} x_i c(s_i) + \frac{2\pi \mathrm{i}}{3} \sum_{i=1}^{5} \lambda_i s_i s_{i+1} \right), \qquad U_i = I + (e^{\mathrm{i}\theta_i} - 1)|+\rangle\langle +|, \qquad |+\rangle = 3^{-1/2} \sum_{s=0}^{2} |s\rangle.$$

The mixer angles θ_i depend on the input. A single forward pass uses $\psi_L = [(\bigotimes_i U_i)D]^L \psi_0$, with the same parameters in each repetition. It measures each local clock $\sum_{\mathbf{s}} |\psi_L(\mathbf{s})|^2 c(s_i)$. The product arm removes the two-wire phase from D. All arms retain the same neighbor-conditioned mixer parameterization. The dephased arm repeatedly applies $P_i(s,t) = |U_i(s,t)|^2$ to each wire's probabilities. The max-plus arm replaces the local sum-product updates by maximum log path scores and normalizes only after the final update. Its joint score separates because the local path scores add.

Depth contrast. The following differences are four minus one entangling layers, in log bp MAE. We average directions and fixed seeds and resample UTC days in 2,500 paired block draws. The intervals are pointwise.

Panel	Windows	Difference	95% UTC-day interval
November afternoon	1,329	−0.001915	[−0.003302, −0.000612]
October	2,048	−0.000364	[−0.001589, +0.000852]
CHF	2,048	+0.008927	[+0.007418, +0.010248]
September stress	2,048	−0.001537	[−0.002569, −0.000572]
Validation	2,048	−0.001391	[−0.002904, +0.000058]

The calendar panels reuse previously scored observations. The validation panel also selects checkpoints, so its contrast is descriptive. The intervals measure variability across the observed days with the two initializations held fixed.

All matched forecast controls

Each entry is the average log bp MAE over both directions and both fixed seeds. The experiment sealed every model's predictor array before decoding the target members. The one-layer product circuit leads across all five panels, although the four-layer entangler has lower MAE than its own one-layer configuration on four panels. On November afternoon, the four-layer entangler minus four-layer product difference is +0.000620 log bp with a paired 95% interval of $[-0.000149, +0.001366]$.

Depth	Propagation	Nov. afternoon	October	CHF	Sept. stress	Validation
1	Entangling	0.175644	0.188882	0.206553	0.187853	0.200536
1	Product	0.168096	0.184369	0.205910	0.183245	0.194749
1	Dephased	0.187957	0.198949	0.228921	0.198410	0.210358
1	Max-plus	0.180552	0.192409	0.214615	0.190803	0.203606
2	Entangling	0.185490	0.197338	0.227055	0.197319	0.208437
2	Product	0.176323	0.188992	0.210342	0.187920	0.199626
2	Dephased	0.195911	0.203979	0.216138	0.203723	0.216278
2	Max-plus	0.183408	0.193389	0.213671	0.191506	0.204583
4	Entangling	0.173729	0.188518	0.215480	0.186316	0.199144
4	Product	0.173110	0.187039	0.206208	0.184707	0.196741
4	Dephased	0.196681	0.206378	0.218516	0.206925	0.219277
4	Max-plus	0.183378	0.193343	0.213551	0.191438	0.204545
–	Persistence	0.196978	0.206430	0.218522	0.207345	0.219583

What the viewer displays. At the same observed training frame from each selected checkpoint, we retain the prepared state and the states after each diagonal and mixer stage. A coherent stage gives the local reduced state $\rho_i(s,t) = \sum_{\mathbf{e}} \psi(s,\mathbf{e})\overline{\psi(t,\mathbf{e})}$, with environmental levels $\mathbf{e}$ summed over the other five wires. The diagonal is the displayed measurement law, while $\mathrm{tr}(\rho_i^2)$ gives its purity. This calculation uses explicit matrix-axis products independent of the optimized production patch. The two displayed forecasts come from the entire input window. The dephased and max-plus traces show their local probability laws and transition matrices.

Exact and numerical verification. An RTX 4090 simulated the 24 fits in 103.81 seconds. Measured peak PyTorch allocations were 482.39 to 568.19 MiB. An independent cuTensorNet contraction [2] compares four trained entanglers (two depths and two seeds) on four observed frames per model. The maximum absolute differences are 4.05×10^{-16} for the 729 complex amplitudes and 2.22×10^{-15} for the checked parameter gradients. Independent NumPy replay reproduced 250 aggregate scores and all 435 paired intervals and rehashed 125 prediction arrays and 72 epoch files. Both GPU methods simulate the circuit classically. Later-date transfer, net returns, and performance on quantum hardware require separate experiments.

Interactive results and circuit inspection Aggregate scores and evidence digests

References

[1] J. Kim et al. Pure Transformers are Powerful Graph Learners. NeurIPS, 2022.

[2] NVIDIA. cuTensorNet contraction API. cuQuantum Python documentation.