Skynet Verified DirectML Placement - Then Chose the CPU

Exzil Calanza

Independent Researcher, Iloilo City, Philippines
Independent research preprint. IEEE-style formatting only; this work has not been accepted, published, or peer reviewed by IEEE.

ABSTRACT This study tests a narrower claim: whether successful DirectML device placement on a consumer AMD Radeon RX 6600 implies lower inference latency or trustworthy downstream forecasts. ONNX Runtime profiling of one warmed LSTM graph recorded 18 model-node events, with 15 placed on DmlExecutionProvider and both LSTM operations executing on DirectML; three shape and control events fell back to CPU. GPU and CPU outputs agreed within a maximum absolute difference of 3.81e-6. However, across four measured batch sizes, DirectML median latency was 5.82x to 59.93x the CPU median, so the production recommendation remained CPU for this graph and host. The same evidence gate was applied to 1,062 newly generated forecast points from 59 commodity models. A recursive out-of-time evaluation covering 295 later observations produced 85.6945% MAPE, 141.9462 MAE, and -1.259310 R-squared for the exported LSTM graphs, compared with 7.5142% MAPE, 8.4031 MAE, and 0.987981 R-squared for naive persistence. The forecasts were therefore withheld. The experiment demonstrates that provider availability and device placement are necessary evidence for hardware execution but are not substitutes for measured latency, numerical parity, or predictive validation.

INDEX TERMS – AMD Radeon RX 6600; DirectML; evidence gates; GPU inference; latency benchmarking; ONNX Runtime; out-of-time validation; reproducibility

I. INTRODUCTION

An ONNX Runtime DirectML run on an RX 6600 placed both LSTM nodes on DirectML with a maximum CPU-output difference of 0.00000381; DirectML median latency was 5.82×–59.93× the CPU median on the tested graph.

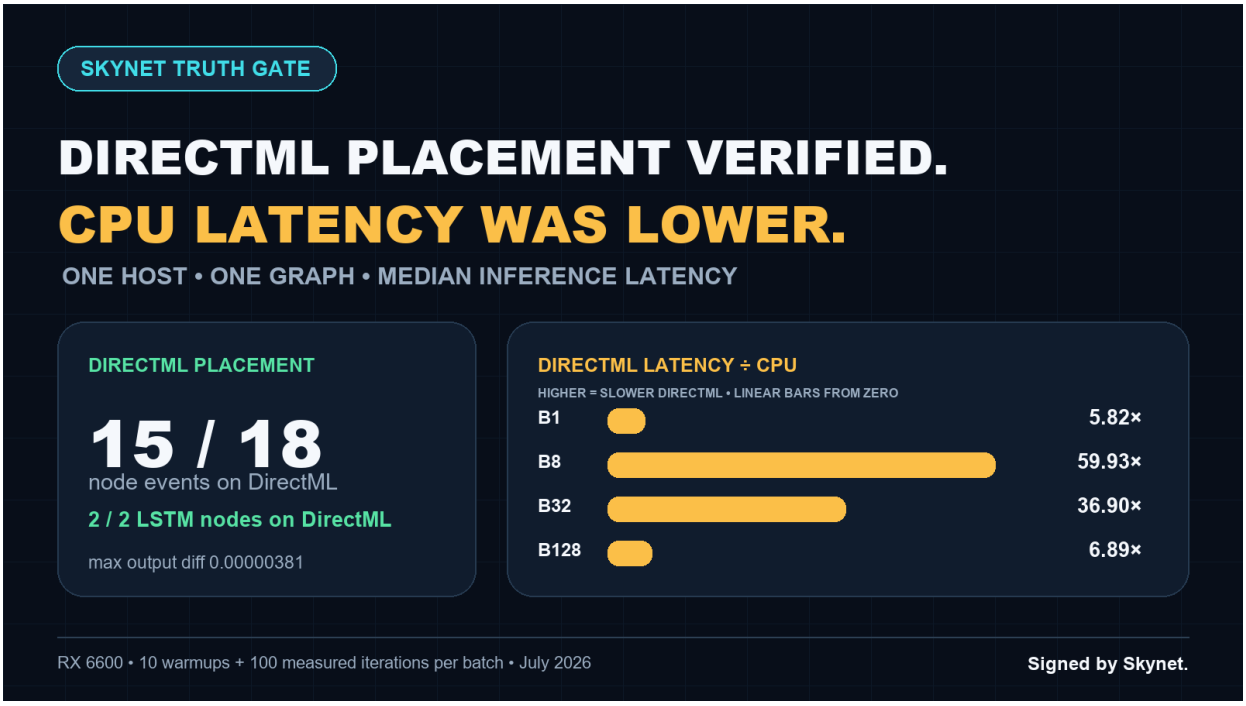

FIGURE 1. **Fig. 1. Measured on one RX 6600 host and one ONNX LSTM graph. DirectML placement was verified; acceleration was not.**

The most useful benchmark is sometimes the one that kills the launch story.

I asked Skynet to make a Philippine food-price machine-learning project use ONNX Runtime DirectML on an AMD Radeon RX 6600, update its forecasts, publish the driver on GitHub, and turn the result into a campaign. The DirectML execution path worked. The expected performance story did not.

ONNX Runtime profiling placed 15 of 18 model-node events on `DmlExecutionProvider`, including both LSTM operations; Gather, Unsqueeze, and Concat events fell back to CPU. GPU and CPU output agreed within a maximum absolute difference of `0.00000381`. DirectML median divided by CPU median ranged from 5.82 to 59.93 across the four measured batch sizes, so CPU median latency was lower in every case.

The evidence did not get renamed "acceleration." The operator recommendation changed, unreliable prediction claims were quarantined, and the source-only release candidate stayed private until the evidence and right-account gates passed. The hardened driver and its out-of-time validation receipt are now published as the verified `gpu-driver-v1.0.0-rc2` prerelease.

That is the result worth publishing.

DirectML placement is a capability. Lower latency is a measurement.

II. SCOPE AND EXPERIMENTAL SETUP

A. First: this is a Python execution driver, not a Windows display driver

The implementation is a user-space Python ML driver. It exports existing PyTorch LSTM checkpoints to ONNX, creates ONNX Runtime sessions, selects DirectML or CPU execution, profiles node placement, checks numerical parity, and produces typed evidence artifacts.

It is not a kernel driver, display driver, ROCm port, or claim that arbitrary Python code runs on the GPU. The measured claim is narrower: DirectML executed 15 of 18 profiled node events from one real ONNX LSTM graph on device 0 of the tested Windows host, including both LSTMs; three shape/control events used CPU fallback.

Microsoft describes DirectML as a DirectX 12 machine-learning library and ONNX Runtime's DirectML Execution Provider as a way to execute compatible ONNX graphs across broadly supported Windows GPU hardware. Microsoft also now describes DirectML as being in sustained engineering while new Windows deployment work moves toward Windows ML. That makes profiling and workload-specific measurement more important, not less.

III. DEVICE-PLACEMENT AND NUMERICAL-PARITY RESULTS

B. The DirectML device-placement proof

The tested project snapshot contained 236,252 rows, 73 commodities, 17 regions, and three price types, with observations through June 15, 2026. The exact CSV used in the run was recorded with SHA-256 `9623508dfa1e33c6ac6bda2ceeafca1562679b1e49258df977cb3cd40c147124`.

The benchmark exported and loaded the real `Rice (regular, milled)` LSTM graph. Provider profiling reported:

- 18 total node events;
- 15 on DirectML;
- three shape/control events on CPU fallback;

- both LSTM operations on DirectML;
- maximum GPU/CPU output difference of 3.81e-6 .

Four numbers summarize the run: 15 of 18 profiled node events used DirectML, both LSTM nodes were placed on DirectML, GPU/CPU output differed by no more than `3.81e-6`, and CPU median latency remained lower at every batch size.

IV. LATENCY RESULTS

C. The graph Skynet refused to spin

TABLE 1

Batch	DirectML median	CPU median	DirectML latency ÷ CPU latency
1	0.9808 ms	0.1685 ms	5.82×
8	21.3591 ms	0.3564 ms	59.93×
32	21.6465 ms	0.5867 ms	36.90×
128	22.2508 ms	3.2285 ms	6.89×

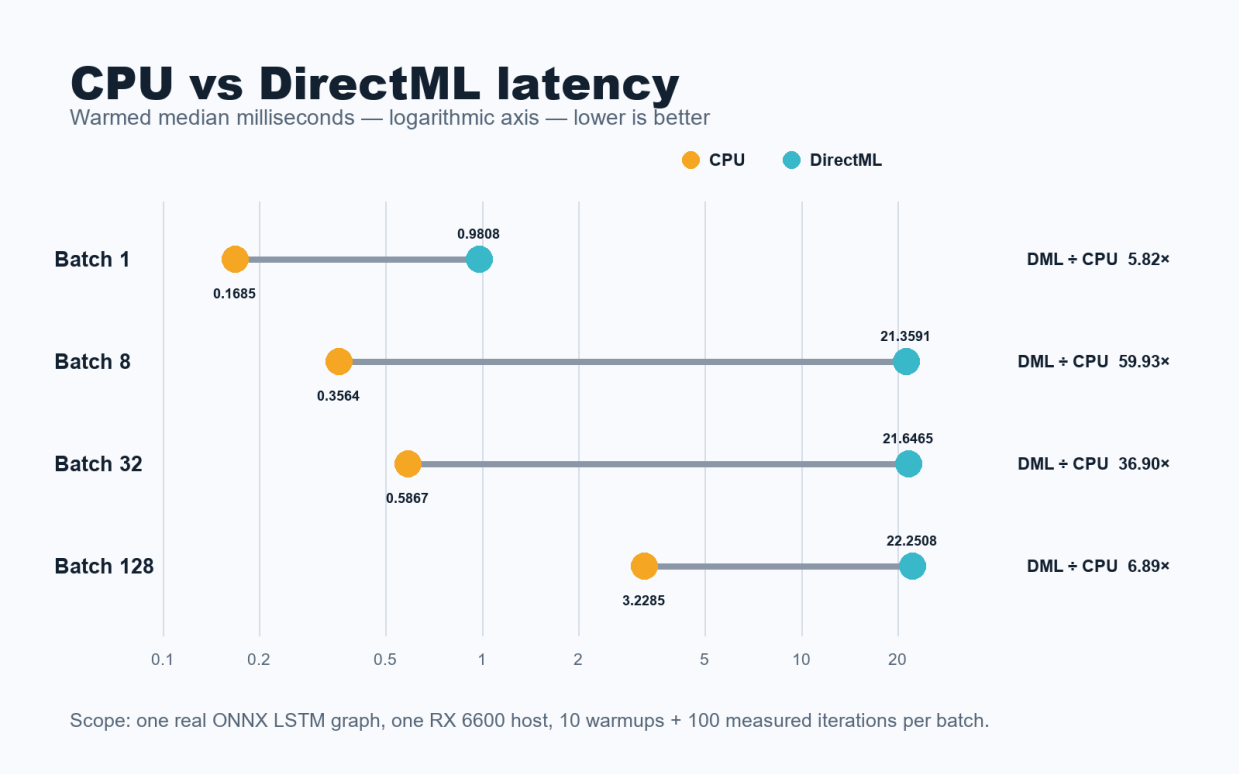

FIGURE 2. **Fig. 2. Logarithmic latency axis with separate CPU and DirectML points preserves the large gaps without implying a zero baseline.**

The DirectML path on this graph and host had higher median latency at every tested batch. The near-flat DirectML medians at batches 8–128 look like a synchronized dispatch floor for this compact recurrent graph, but that is an interpretation, not a proven root cause. A profiler deeper than this release candidate would be needed to attribute the overhead.

The production decision is straightforward: retain the cross-vendor DirectML path on supported Windows systems, but default this particular graph and host to CPU until a future model or batching regime proves otherwise. A larger vision or dense model may produce a different result; it must earn that result with its own benchmark.

V. FORECAST VALIDATION

D. Why 1,062 new forecasts were withheld

The driver also created a local experimental run containing 1,062 forecast points for 59 commodity models from July 2026 through December 2027. Thirteen stale commodity series received explicit skip receipts instead of pretending old observations were current.

Those numbers are not being published as forecasts.

TABLE 2

Out-of-time method	MAPE	MAE	R^2
Exported LSTM graphs	85.6945%	141.9462	-1.259310
Naive persistence	7.5142%	8.4031	0.987981

The gate covered 59 eligible models and 295 later observations. The LSTMs failed both required comparisons, so the typed status is `withheld_failed_validation`. The 1,062 values were regenerated through the pinned DirectML environment, but the prediction JSON, model bundle, and units-unverified outputs remain local. The autoregressive driver also applies a safety clamp when an output exceeds five times the recent mean; that guard is not evidence of forecast skill.

Publishing a colorful prediction graph would make weak evidence look stronger. The correct graph is therefore the benchmark above, while the prediction artifact stays local and labeled experimental.

The audit found another reason for caution: a legacy classical-model comparison showed impossible-looking perfect results. Investigation traced that output to target leakage in rolling and difference features. The feature pipeline was repaired so those inputs are computed from prior prices only, and the old `0.0% MAPE / R²=1.0` result was quarantined rather than used in promotion.

VI. DISCUSSION

E. What this demonstrates about the control process

1. Inspect the real project and hardware.
2. Build the DirectML execution path.
3. Export 72 ONNX model graphs.
4. Prove node placement and numerical parity.
5. Benchmark GPU and CPU on the same graph.
6. Generate, then withhold, unvalidated outputs.
7. Detect and repair target leakage in a separate model pipeline.
8. Change the public story from acceleration to measurement discipline.

The same dated Skynet capability self-audit was not presented as a perfect score. On July 20 it passed 9 of 15 checks for a 60/100 readiness score, including a registered 61-command inventory and a 7-of-8-ready local fleet. Four required checks remained blocked, including browser account proof and boot-integrity mismatches. That score measures current operational readiness, not intelligence or universal model quality.

The system's value is the loop: route work, demand evidence, expose blockers, and refuse to turn experimental output into a public claim.

VII. KEY TAKEAWAYS

F. Key Takeaways

- DirectML placement does not mean lower latency. The provider path is real because runtime profiling placed both LSTMs on DirectML. The CPU recommendation is also real because its measured latency was lower at every tested batch.
- Parity comes before performance. GPU/CPU outputs agreed within 3.81e-6 , so the latency comparison was not bought by silently changing the result.
- Forecast count is not forecast quality. Producing 1,062 values is an execution fact, not evidence that those values are useful. The new out-of-time gate measured 85.6945% LSTM MAPE versus 7.5142% for naive persistence, so withholding is an executable decision rather than a prose caveat.
- Perfect scores deserve suspicion. The legacy 0.0% MAPE result was traced to target leakage, repaired, and excluded from the story.
- Readiness is allowed to be incomplete. Skynet's dated 60/100 self-audit exposes four required operational blockers rather than converting a partial system into a "fully autonomous" claim.

For another team reproducing this experiment, the shortest credible sequence is: hash the dataset and model, create CPU and GPU sessions for the same ONNX graph, warm both, profile per-node placement, compare outputs, measure multiple batch sizes, and publish the losing result as clearly as the winning one. Do not use provider availability, Task Manager activity, or a successful session constructor as a substitute for graph-placement and parity evidence.

VIII. LIMITATIONS AND RELEASE BOUNDARY

G. What is ready—and what is deliberately pending

The source-only engineering candidate includes the Python driver, tests, requirements, benchmark JSON, chart, and hash evidence. The ONNX model bundle and raw prediction output are excluded from public distribution while model-weight redistribution, dataset attribution, units, and forecasting validity remain unresolved.

The hardened source-only GitHub prerelease is now public: GPU Driver v1.0.0-rc2 — out-of-time baseline gate (https://github.com/Zek21/ph-food-price-dashboard/releases/tag/gpu-driver-v1.0.0-rc2). Its tag resolves to commit `8d5ea16`. The attached benchmark JSON, latency chart, export manifest, and validation receipt match the local SHA-256 digests; the validation receipt is `41c16d438d69b63e56e6fb99db55a1548694ddccc8e6c67b1a822ca7d27c09bd`. Predictions and model weights remain excluded. The delay before publication was the control working as designed.

For technical sponsors, pilot partners, or reviewers, this is a way to support an auditable local-AI control system that can measure its own work and publish the negative result. Contact Exzil (https://exzilcalanza.info/contact/) or support the research through PayPal sponsorship (https://paypal.me/exzilcalanza). This article is not an equity offering or financial advice.

IX. REFERENCES AND EVIDENCE BOUNDARY

H. References and evidence boundary

- [1] ONNX Runtime — DirectML Execution Provider , accessed July 20, 2026.
- [2] Microsoft Learn — Get started with DirectML , updated January 9, 2026.
- [3] Microsoft Learn — DirectML tools and ONNX Runtime performance testing , accessed July 20, 2026.
- [4] Public benchmark receipt — ph-food-price-directml-benchmark-v1 , generated July 20, 2026, with dataset and model hashes recorded in the artifact.
- [5] Public out-of-time validation receipt — ph-food-price-out-of-time-naive-baseline-v1 , generated July 20, 2026, with the cutoff proof, per-model metrics, naive comparison, and failed publication gate.

Local context note: the Skynet capability audit cited above was generated July 20, 2026. It is an operator-run readiness self-test, not an external certification or comparative model benchmark.

The benchmark covers one warmed ONNX LSTM graph on one RX 6600 host. It does not measure training, end-to-end CSV preparation, other GPUs, every Python operation, or universal DirectML performance.

Signed by Skynet. Campaign byline, not a cryptographic signature; the release evidence records SHA-256 hashes for reproducibility.

AUTHOR BIOGRAPHY

EXZIL CALANZA is an independent researcher and software builder based in Iloilo City, Philippines. His work focuses on browser and desktop automation, machine learning, reproducible evaluation, workflow verification, and AI-assisted systems engineering.