# Backend-Dependent Stability on Legacy HBM GPUs

*ROCm/HIP vs Vulkan on Vega/gfx900: residual paths, offload-layer semantics, and reproducible diagnosis*

Akira Ito | AETS (Akatsuki Enterprise Technology Solutions) | aets-giken@hiroshima-aktk.com

IEICE GNW-68 | Kyushu Sangyo University | March 9, 2026

## 1 Background & Motivation

- Official support matrices not listed gfx900 in ROCm 7.2, but real systems can still execute through residual code and artifact paths.
- For students and small labs, the key question is whether execution is possible, stable, and reproducible.
- We built a matched dual-backend testbed on the same Vega device to separate support policy, backend choice, and runtime behavior.
- **Goal: explain why "unsupported" and "unusable" are not the same statement.**

## 2 Problem Statement

- Legacy Vega GPUs are often treated as categorically unsuitable for modern ROCm workloads.
- Success or failure may be decided at different layers: distribution presets, build targets, init validation, backend macros, or runtime compute paths.
- **We ask: is instability caused by Vega itself, or by a specific backend path under matched conditions?**

## 3 Confirmed Facts from Investigation

### A. gfx900 Gate Matrix (repo: work_log/investigations/gfx900_gate_matrix.md)

| Layer | Current handling of gfx900 | Blocks / Allows |
|---|---|---|
| **Official matrix (ROCm 7.2)** | gfx900 not listed in GPU target list | **Blocks (official scope)** |
| **CMake filter (ollama)** | Default regex excludes gfx900; manual override possible via AMDGPU_TARGETS | **Blocks default / Allows manual** |
| **rocBLAS artifacts** | Kernels.so-000-gfx900.hsaco found in /usr/lib/ollama/rocblas/library/ | **Allows** |
| **Runner init validation** | Deep init probe (GGML_CUDA_INIT=1): gfx900 recognized as valid agent | **Allows** |
| **Source macros (hip.h, common.cuh)** | __gfx900__ → GCN5 classification retained; dp4a path exists | **Allows** |
| **Runtime execution** | ROCm/HIP: all num_gpu values OK. Vulkan: SIGSEGV when num_gpu>=1 | **Backend-dependent** |

### B. num_gpu Semantics Trace (repo: work_log/investigations/numgpu_semantics_trace.md)

| Stage | Code location | Meaning |
|---|---|---|
| **Python client** | ollama-python/_types.py:109 | Load-time option |
| **Ollama CLI** | cmd/interactive.go:112 | **"number of layers sent to GPU"** |
| **Server assign** | llm/server.go:992,1063 | requestedLayers = NumGPU |
| **Runner bridge** | llama/llama.go:266 | cparams.n_gpu_layers |
| **llama.cpp** | include/llama.h:289 | **"layers stored in VRAM"** |

**Conclusion:** num_gpu is offloaded layer count, NOT GPU device count. num_gpu>=2 does not imply multi-GPU.

### C. Related Literature (cited, not reproduced in this repo)

- LLNL differential testing (arXiv:2410.09172): 600K+ auto-generated kernels showed FP32 divergence ~9% between NVIDIA V100 and AMD MI250. Root causes: vendor-specific math libraries, compiler flags, HIPIFY artifacts. Motivates our backend-aware validation approach.
- gem5 Vega modeling (Univ. Wisconsin-Madison): GPUFS mode runs real Linux kernel + unmodified amdkfd driver. Validated against physical Vega 56 with avg 6% error. Establishes Vega as academically credible.
- ROCm SDMA workarounds (AMD docs + LUMI reports): HSA_ENABLE_SDMA=0 bypasses faulty DMA with Blit kernel fallback (~80% bandwidth). HSA_OVERRIDE_GFX_VERSION enables unsupported GPU execution — the override used in our experiments.
- **These are external references. All claims in this poster are based solely on repo-internal evidence (Sections 3A, 3B, 6).**

## 4 Engineering Challenges

- gfx900 excluded by official matrices and default build filters, yet residual code and artifacts still retain gfx900 handling.
- **`num_gpu` is easily misread as GPU count; our code trace confirms it means offloaded layers.**
- The same Vega GPU can succeed on ROCm/HIP and fail on Vulkan under the same model and prompt.
- Diagnosis must be layer-by-layer and evidence-driven, not reduced to "legacy GPU bad".

## 5 Evidence-First Investigation Strategy

### Investigation Layers

| Layer | Intervention | Effect |
|---|---|---|
| **L1: Distribution / Build** | Inspect support matrix, presets, target filters, installed artifacts | Locate where gfx900 is blocked, allowed, or only partially retained |
| **L2: API Semantics** | Trace num_gpu through client → server → runner → llama.cpp | Fix interpretation: offloaded layers, not GPU count |
| **L3: Runtime Comparison** | Run matched ROCm (:11435) and Vulkan (:11434) tests | Isolate backend-specific failure modes on identical hardware |
| **L4: Evidence Capture** | Structured work logs, journal traces, backend probes, rocm-smi, ollama ps | Localize crashes and make claims reproducible |

*Approach prioritizes falsifiable diagnosis and reproducibility over optimistic one-off success.*

## 6 Experimental Results

### Test Environment

| Component | Specification |
|---|---|
| **GPU** | AMD Radeon RX Vega 56 (gfx900), 8 GB HBM2 |
| **OS / Kernel** | EndeavourOS, Kernel 6.18.16-1-lts |
| **ROCm path** | Ollama 0.17.5 via :11435, library=ROCm, libggml-hip.so |
| **Vulkan path** | Ollama 0.17.4 via :11434, library=Vulkan, same model/prompt |
| **Override** | **HSA_OVERRIDE_GFX_VERSION=9.0.0 (ROCm service only)** |
| **Note** | Vulkan=0.17.4 vs ROCm=0.17.5: version difference is inherent to the dual-install; both tested as-shipped |

### Matched-Condition Results (qwen3.5:2b, NUM_PREDICT=512)

| num_gpu | ROCm (:11435) run_20260307_012643 | Vulkan (:11434) run_20260307_013050 |
|---|---|---|
| **0 (CPU only)** | **OK (46.7s, eval_count=512)** | **OK** |
| **1** | **OK (48.7s, eval_count=512)** | **FAIL (HTTP 500, SIGSEGV)** |
| **2** | **OK (47.7s, eval_count=512)** | **FAIL (HTTP 500, SIGSEGV)** |
| **-1 (all layers)** | **OK (44.3s, eval_count=512)** | **FAIL (HTTP 500, SIGSEGV)** |

**Note:** Vulkan succeeded with num_gpu=0 (no GPU offload). Failures occurred only when num_gpu>=1 (GPU compute path active). ROCm results are within this tested scope (single model, short fixed runs); broader workloads may differ. Model dependency observed: tinyllama succeeded on Vulkan in earlier tests (run_20260307_003423, 20/20 OK); qwen3.5:2b triggers the SIGSEGV. Also: num_gpu=0 runs may produce eval_count=512 with response_chars=0 — this is a model output quirk, not a crash.

**Crash localization:** Vulkan SIGSEGV occurred in ggml_backend_sched_graph_compute_async → computeBatch, after model load completed and runner started. This is a compute-phase failure, not an initialization failure.

### Failure Diagnosis

| Hypothesis | Expected Evidence | Observed |
|---|---|---|
| Vega is generally unusable | Both backends fail on same GPU | **Contradicted: ROCm/HIP succeeded** |
| num_gpu means GPU count | Failure implies multi-GPU entry | **Contradicted: code trace = offloaded layers** |
| **Vulkan compute-path instability** | Load OK; crash after runner start in compute stack | **MATCH** |

### Why gfx900 Still Works Sometimes

ROCm changelog says "no longer built by default" rather than "forbidden." In the local install, gfx900 hsaco artifacts and libggml-hip.so strings were verified present.
Interpretation:
**unsupported = unguaranteed and untested, not necessarily impossible.**

### Limitations & Future Work

**Limitations:** Matched comparison used short fixed runs (1 epoch, 512 tokens) with a single model (qwen3.5:2b). Vulkan and ROCm Ollama versions differ (0.17.4 vs 0.17.5). tinyllama behaved differently from qwen3.5 on Vulkan, indicating model-dependent failure conditions. Results should not be generalized beyond this tested scope without further verification.

**Future work:**
Multi-epoch stress tests, additional models (llm-jp, phi-3), ROCm version comparison (6.x vs 7.x), and GGML_CUDA_NO_PEER_COPY experiments to isolate offload-path failure mechanisms. Reproducibility infrastructure (workaround matrix with 15 documented cases) is available in the repo for community use.

## 7 Takeaway & Key Messages

### ✅ Support Status ≠ Execution Reality

Official exclusion does not imply impossibility. Gate matrix shows 5 of 6 layers allow gfx900 execution. Residual code paths and shipped artifacts make partial execution possible.

### Backend Choice Dominates

On the same Vega device with qwen3.5:2b, ROCm/HIP passed all num_gpu conditions within tested scope. Vulkan failed when num_gpu>=1 (GPU offload active). Failure is backend-specific, not architecture-wide. Model dependency (tinyllama vs qwen) exists.

### Reproducibility Before Claims

All claims backed by specific run_ids and code traces. Key runs: run_20260307_012643 (ROCm qwen), run_20260307_013050 (Vulkan qwen), run_20260307_003423 (tinyllama 20/20 OK). Gate matrix and workaround matrix (15 cases) in repo.

***Unsupported ≠ Impossible. | Legacy ≠ Poor. | Backend choice matters.***