# When Smaller Is Slower: Dimensional Collapse in Compressed LLMs

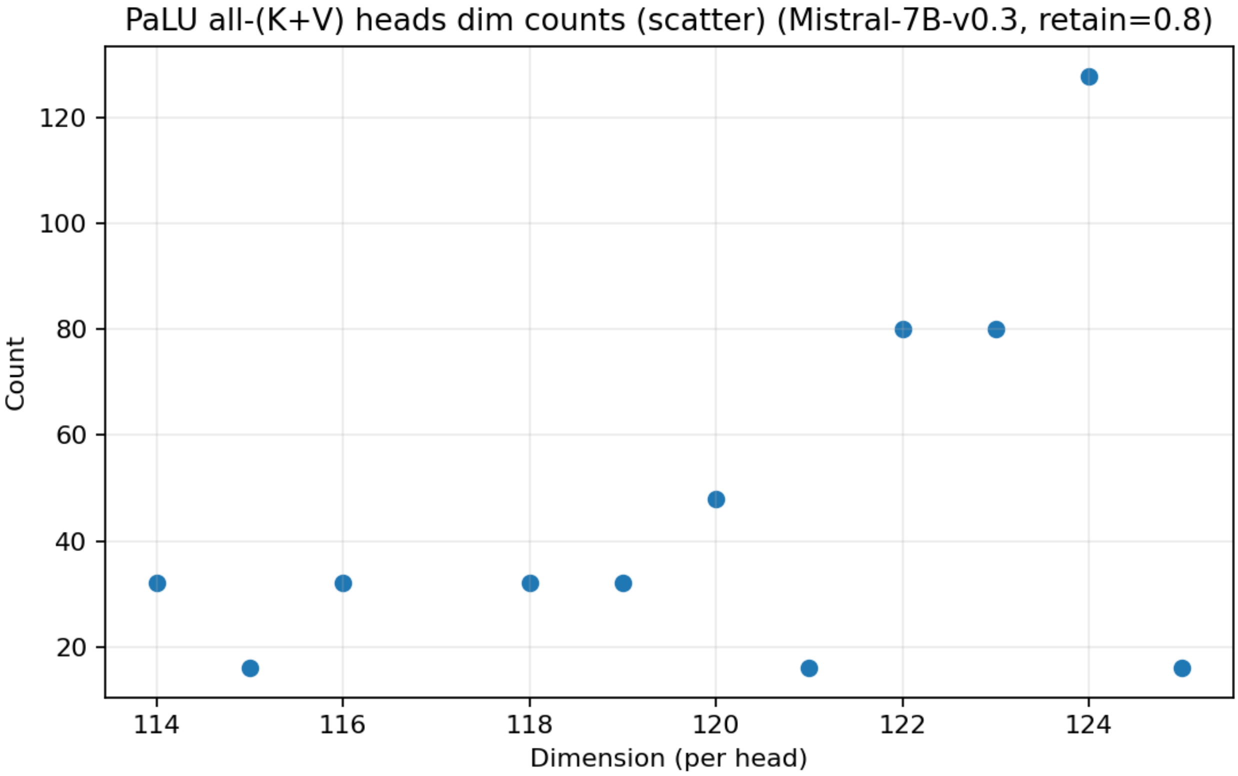
Suggested alternatives: **ShapeGuard: Preventing Dimensional Collapse in Compressed LLMs**; **GAC: GPU-Aligned Compression for Fast LLM Inference**.

## Motivation

**Post-training compression reshapes LLM operators (e.g., per-head dimensions become heterogeneous and irregular).
Irregular dimensions can trigger nonlinear GPU slowdowns ("dimensional collapse"), even when FLOPs decrease.
Kernel-aligned constraints (pad/pack/round) can rescue performance with small memory overhead.**

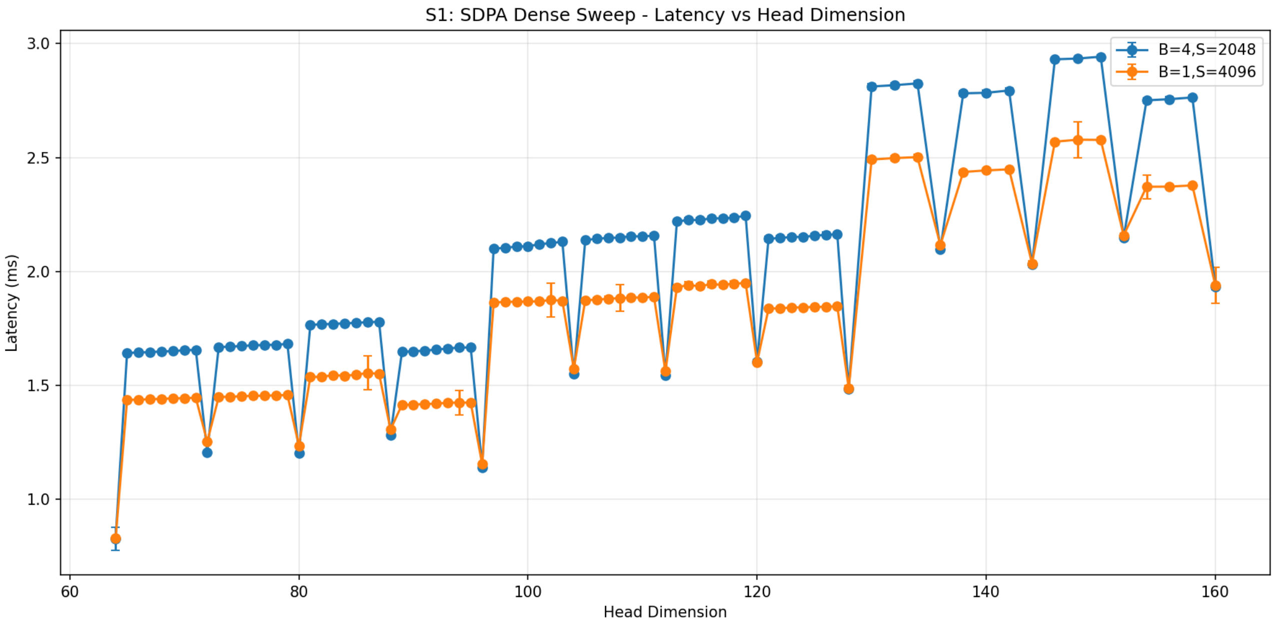
# 1. Dimensional Collapse

## 1.1 Compression changes per-head dimensions




Example: PaLU dimension distribution for Mistral (retain=0.8); per-head K/V dims vary.

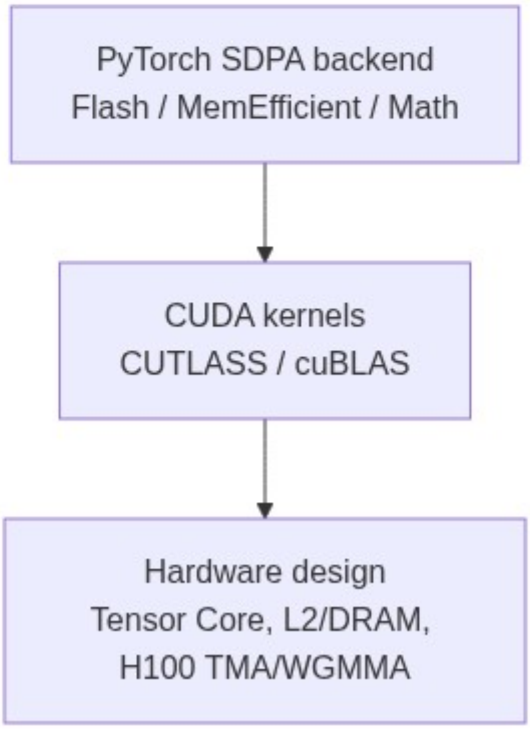
## 1.2 SDPA latency shows alignment cliffs



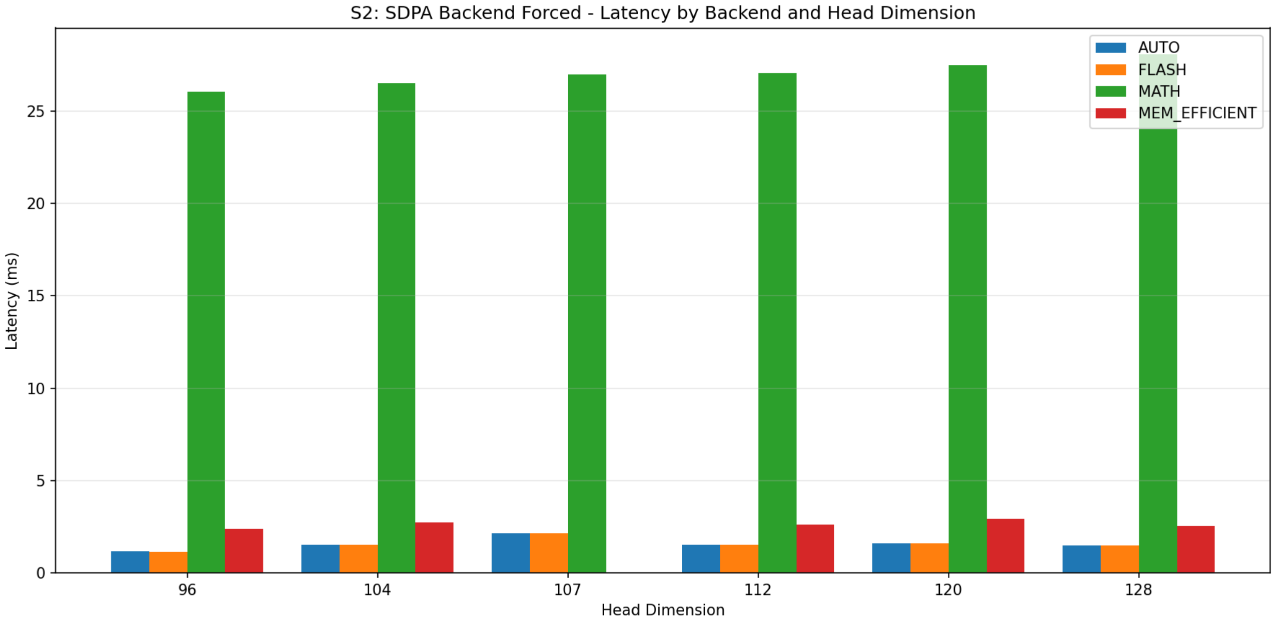
**Experiment config (S1 / A100)**: `B=4,S=2048,H=32`, dtype=`fp16`, `is_causal=False`, warmup=50, measure=200, trials=3.
**Key points**: `D=96: 1.140 ms`, `D=107: 2.147 ms` (**+88%** vs 96), `D=112: 1.545 ms` (~**-28%** vs 107).

---

## 2. Possible Cause (Hypotheses)



## 2.1 PyTorch backend selection (evidence)




**Experiment config (S2 / A100)**: `B=4,S=2048,H=32`, dtype=`fp16`, `is_causal=False`, head_dim ∈ {96,104,107,112,120,128}.
**Observed at `D=107`**: `FLASH≈2.139 ms`, `MATH≈26.995 ms` (**~12.6×**); `AUTO≈FLASH`; `MEM_EFFICIENT` unsupported in this run.

## 2.2 CUDA kernel layer (hypotheses)

- Kernel variants are heavily tiled/vectorized; irregular `D` can force predication, worse memory coalescing, or less-optimized kernels.
- CUTLASS/cuBLAS GEMM often prefers aligned `K/N` (Tensor Core tiles); irregular dims may trigger slower paths or extra packing.

## 2.3 Hardware layer (hypotheses)

- Tensor Cores prefer tile-compatible dimensions (often `K % 16 == 0` for FP16/BF16).
- L2/DRAM transaction granularity makes misaligned row strides waste bandwidth.
- On H100, TMA/WGMMA introduce stricter alignment/stride contracts; irregular strides can disable the fastest data-movement/compute paths.

---

# EuroMLSys

We frame this project as a **new post-training compression paradigm**: compression must satisfy a GPU “shape contract”, otherwise irregular operator shapes can trigger dimensional collapse.

## C1) Quantify the phenomenon

## C2) How to probe the system

## C3) Formulate as a constraint optimization problem

## C4) Solver + repair pass (optional)

## C5) Kernel Level Validation