# Meet Banthia

(+1) 551 353 6752 | mjb9520@nyu.edu | meetbanthia | meetbanthia | Personal Website

## Education

**NYU, Courant Institute of Mathematical Sciences** — **Aug 2025 – May 2027**
*Master of Science (with Thesis) in* ***Computer Science*** — ***GPA: 3.93/4.0***

**NIT Karnataka, Surathkal** — **Dec 2021 – May 2025**
*Bachelor of Technology in* ***Computer Science and Engineering*** — ***GPA: 8.51/10***

## Publications

- **FastGRT: Accelerating Training of 3D Gaussian Ray Tracing** — **M. Banthia**, X. Liu, H. Zhao, A. Panda, J. Li. Under review at **MLSys 2027** (Conference on Machine Learning and Systems)

## Experience

**NYU MLSys Research Group - *Advisor: Prof. Jinyang Li*** — **Jan 2026 – Present**

**FastGRT: Accelerating Training of 3D Gaussian Ray Tracing *[GitHub]***

***Skills: Kernel/Memory Optimization, CUDA, Vulkan, OptiX, GPU Profiling, Nsight Systems/Compute***

- Built and open-sourced **FastGRT**, the first **Vulkan** implementation of 3D Gaussian Ray Tracing supporting both training and rendering, cutting end-to-end training time **1.9–4.97**× versus NVIDIA's OptiX-based 3DGRT across Mip-NeRF360, Tanks and Temples, Deep Blending and Zip-NeRF at matched PSNR, SSIM, LPIPS and render speed.
- Profiled 3DGRT across scenes and image resolutions, tracing training cost to backward-pass ray re-traversal, contended global gradient atomics and per-iteration BVH rebuilds rather than to traversal itself.
- Designed **adaptive multi-round traversal** that decouples traversal from blending. The forward pass archives every hit per pixel in a coalesced layout under a round budget that never truncates long rays, so the backward pass replays recorded hits instead of re-traversing the BVH, preserving 3DGRT's exact rendering formulation.
- Built a **subgroup-aggregated gradient accumulator** staging 59 gradient components per lane in ∼14.75 KiB of shared memory per 64-invocation workgroup, collapsing global atomics for a **2–10**× **faster backward pass**.
- Added a **hybrid BVH** strategy refitting in place and rebuilding only when densification or pruning changes the Gaussian population, and moved Gaussian IDs to global memory to raise the optimal **payload buffer size** from 11 to 24.

**Oracle - Server Technology Intern** — **May 2024 – Jul 2024**

- Designed and implemented a Continuous Integration/Continuous Deployment pipeline serving as the quality gatekeeper for Cloud Platform Services, with automated triggers for service releases from active branches.
- Integrated automated test execution on Preflight (Gamma stage) environments where services were deployed in real-time and established a 95% minimum pass rate threshold for integrated tests before upstream promotion to Fusion Apps.

## Projects

**ForkAndMove: Parallel Chess Engine *[GitHub]*** *(team of 2)* — **Oct 2025 – Dec 2025**

- Chess engine on a **bitboard** representation with **bitwise** move generation, evaluation and rule handling (castling, en passant), searched by a **multithreaded parallel Alpha-Beta** with **Principal Variation Search**. The leftmost branch runs sequentially, and sibling subtrees then expand in parallel under the resulting bounds.
- Cut explored states with **move-ordering heuristics** and lazy game-tree expansion, reaching **2.8**× speedup on 4 processors over the single-processor run, **1.9**× over sequential alpha-beta in chess and **3.8**× in Ultimate Tic Tac Toe.

**ParJSON: Parallel JSON Parser in OpenMP *[GitHub]*** — **Mar 2026 – Apr 2026**

- Stage-1 JSON structural scanner in **OpenMP** that handles escaped quotes and backslash runs. Each chunk simulates all four boundary states, and a **compact encoding** replaces full per-state masks to cut redundant writes.
- Benchmarked all-state simulation, compact chunk encoding and a **parallel-scan** formulation, reaching **4.7**× on string masking at 64 threads and up to **3**× overall, verified against Python's `json` module.

## Technical Skills

**Languages**: C/C++, Python, Standard ML, Elixir
**Systems and ML**: CUDA, Triton, Vulkan, OptiX, Nsight Systems/Compute, OpenMP, PyTorch, TensorFlow, NumPy
**Coursework**: **Efficient AI**, **ML Systems**, Distributed Systems, Parallel Algorithms, **Honors Algorithms**

## Teaching

**Course Assistant: Operating Systems (Graduate), NYU Courant** — **Summer'26, Fall'26**

- Co-designed a file-system assignment with **Prof. Hubertus Franke**. An `LD_PRELOAD` shim redirects file-system calls to a fixed-size file students implement as a virtual disk. Office hours, discussion sections and grading.