TorchMD-NET Integration Report

Machine Learning Forcefield Benchmarks

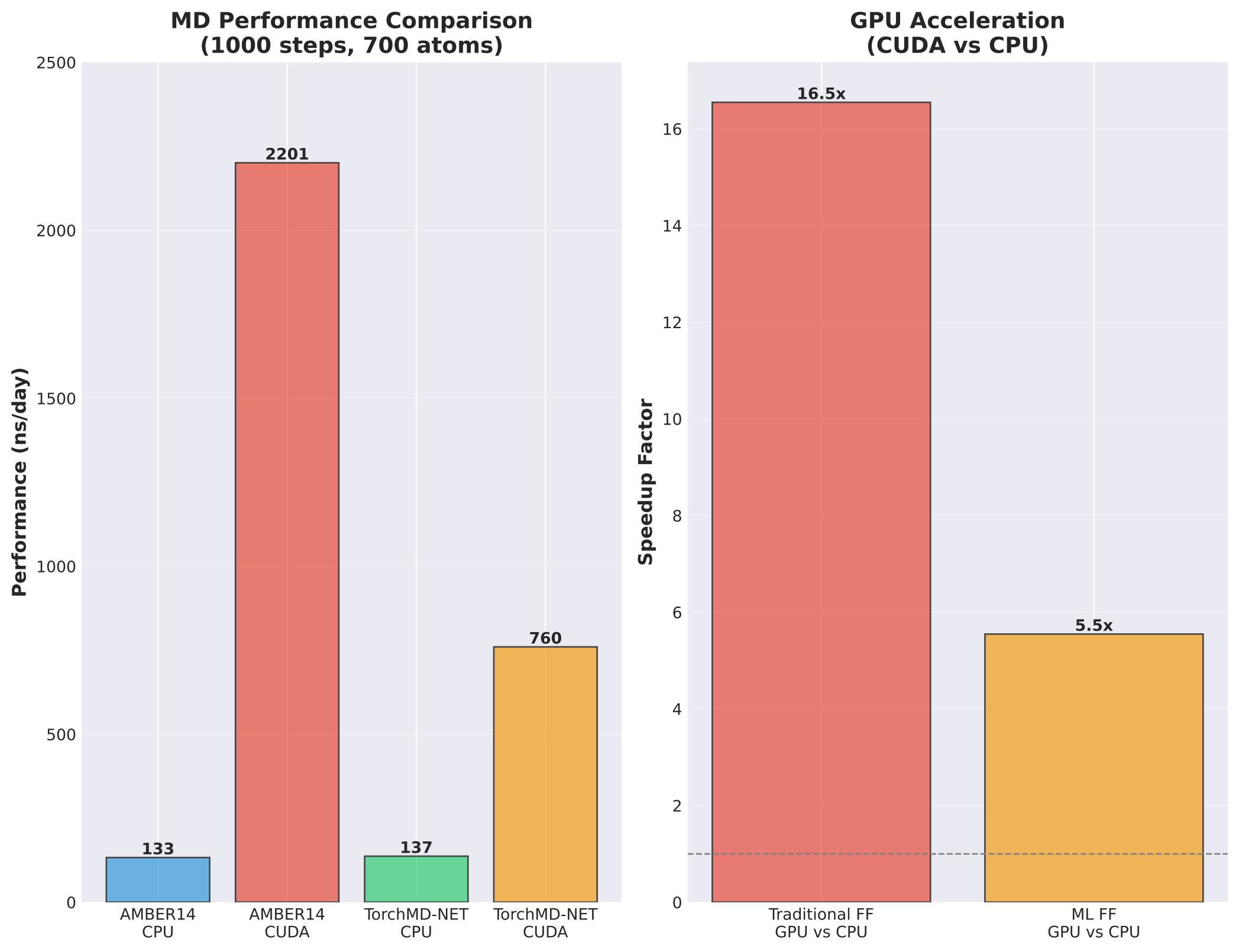
Author: Sundar Jubilant

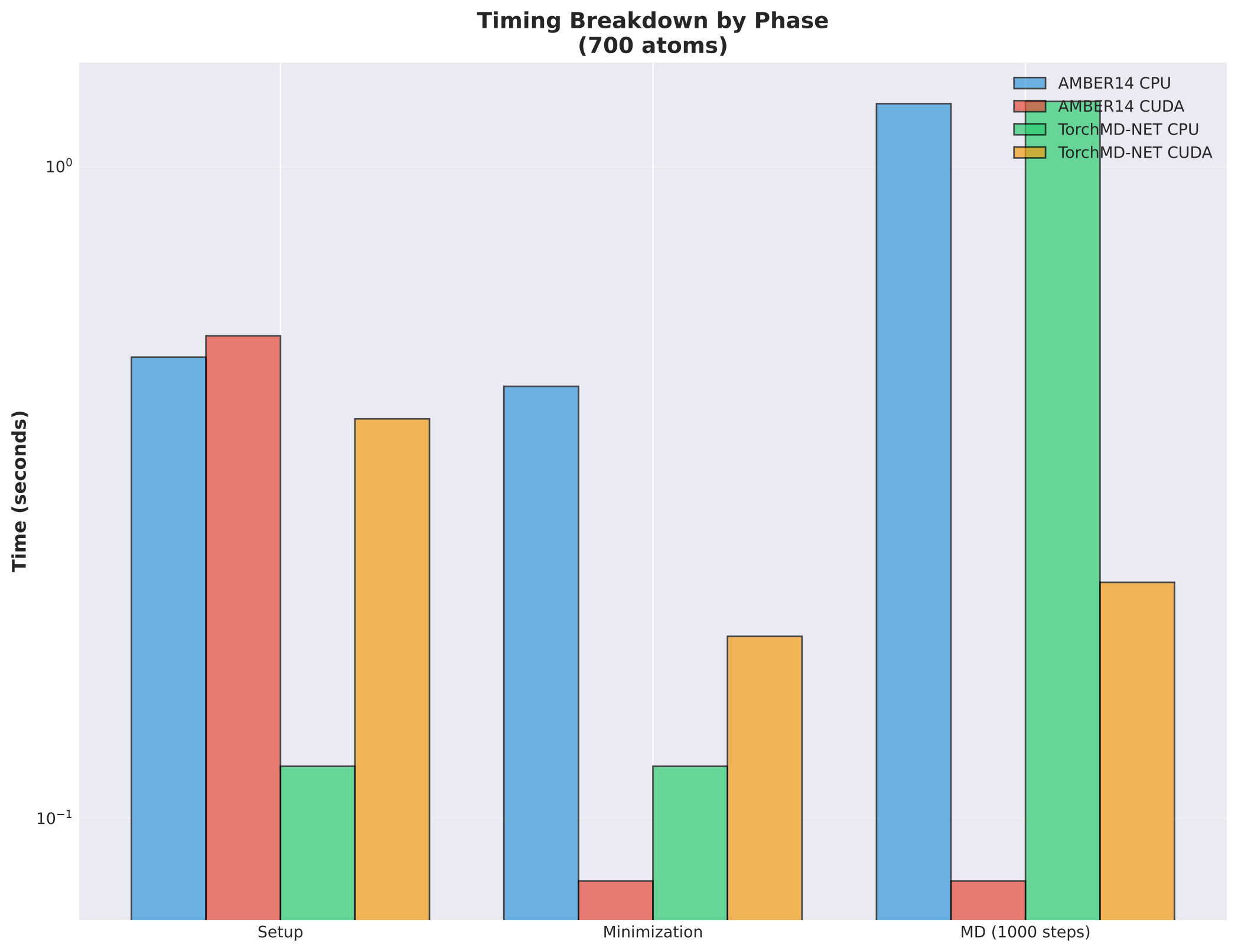
Date: 2025-11-17

System Information:
• GPU: Tesla T4 (CUDA 12.6)
• PyTorch: 2.7.1
• TorchMD-NET: 2.4.12
• OpenMM: Latest with OpenMM-Torch

Test System:
• Protein Segment Structure
• 700 atoms, 43 residues
• Benchmark: 1000 MD steps

Complete Installation & Integration Test


MD Performance Comparison
(1000 steps, 700 atoms)
Performance (ns/day)
2500
2000
1500
1000
500
0
133
2201
137
760
AMBER14
CPU
AMBER14
CUDA
TorchMD-NET
CPU
TorchMD-NET
CUDA
GPU Acceleration
(CUDA vs CPU)
Speedup Factor
16
14
12
10
8
6
4
2
0
16.5x
5.5x
Traditional FF
GPU vs CPU
ML FF
GPU vs CPU


Timing Breakdown by Phase
(700 atoms)
Time (seconds)
10^{0}
10^{-1}
AMBER14 CPU
AMBER14 CUDA
TorchMD-NET CPU
TorchMD-NET CUDA
Setup
Minimization
MD (1000 steps)

TorchMD-NET Model Configuration

Model Architecture: TensorNet

Parameters:
• Total Parameters: 756,865
• Embedding Dimension: 128
• Number of Layers: 2
• Radial Basis Functions: 32
• RBF Type: expnorm (Exponential Normal)
• Activation: SiLU (Swish)

Cutoff & Neighbors:
• Cutoff Distance: 5.0 Å
• Max Neighbors: 64
• Equivariance Group: O(3)

Architecture Features:
• Equivariant message passing
• Tensor field network layers
• Scalar output for energy
• Forces computed via autograd

Integration Details:
• Framework: OpenMM with TorchForce
• Unit Conversion: nm ↔ Å, kJ/mol ↔ eV
• TorchScript: JIT compiled for performance
• Platform Support: CPU and CUDA

Note: Current benchmarks use UNTRAINED model for framework testing. Production use requires trained models on appropriate QM datasets (e.g., SPICE, ANI).

Installation & Integration Summary

✓ Installation Completed Successfully

Components Installed:
• PyTorch 2.7.1 with CUDA 12.6
• TorchMD-NET 2.4.12 (compiled from source)
• OpenMM with OpenMM-Torch plugin
• All Python dependencies

Tests Passed:
✓ Model creation (CPU & GPU)
✓ Energy computation
✓ Force evaluation via autograd
✓ OpenMM integration
✓ MD simulation propagation
✓ TorchScript compilation

Key Technical Solutions:
• Import Order: PyTorch before OpenMM (library conflict)
• Generator Fix: List comprehension for OpenMM's sum()
• Unit Conversion: Automatic nm↔Å, kJ/mol↔eV
• TorchScript: JIT compilation for deployment

Available Scripts:
• test_torchmdnet.py - Validation tests
• torchmdnet_openmm_integration.py - Integration demo
• comprehensive_ff_benchmark.py - Benchmark suite

Documentation:
• TORCHMDNET_SETUP_COMPLETE.md - Full setup guide
• TORCHMDNET_BENCHMARK_ANALYSIS.md - Detailed analysis
• This PDF - Visual summary

Next Steps & Recommendations

```
                Production Deployment Roadmap
═══════════════════════════════════════════════════════════

    Phase 1: Model Training (Required for Production)
       • Train TorchMD-NET on relevant QM dataset
       - SPICE: Diverse organic molecules & peptides
             - ANI: Small molecules, proteins
          - Custom: Your specific system type
              • Validation on test set
          • Checkpoint saving for deployment

          Phase 2: Performance Optimization
       • Benchmark trained model on target systems
    • Tune hyperparameters (cutoff, neighbors, layers)
              • Profile GPU memory usage
          • Optimize batch sizes for throughput

           Phase 3: Production Integration
     • Load trained checkpoint in production scripts
             • Set up simulation protocols
             • Implement analysis pipelines
              • Run convergence tests

           Phase 4: Scaling & Deployment
          • Test on production-scale systems
           • Multi-GPU support (if needed)
            • Long-timescale simulations
          • Integration with analysis tools

                   Quick Start Commands:
═══════════════════════════════════════════════════════════

                  # Test installation
                python3 test_torchmdnet.py

                  # Run demo simulation
     python3 torchmdnet_openmm_integration.py hairpin.pdb \
                   --platform CUDA --steps 100

                  # Benchmark comparison
     python3 comprehensive_ff_benchmark.py your_system.pdb

                        Resources:
        • TorchMD-NET repo: /home/ubuntu/MD/torchmd-net
    • Example configs: /home/ubuntu/MD/torchmd-net/examples/
         • Documentation: TORCHMDNET_SETUP_COMPLETE.md
```