# Bo Pang

Atlanta, GA | +1-404397-3308 | bopang314@gmail.com | LinkedIn | GitHub

## EDUCATION

**Georgia Institute of Technology** *M.S. in Computer Engineering* **GPA: 3.88/4.0** **Atlanta, GA** Aug 2023 – May 2026

**University of Illinois Urbana-Champaign** *B.S. in Computer Engineering* **GPA: 3.82/4.0** **Champaign, IL** Sep 2019 – Jun 2023

## TECHNICAL SKILL

- **AI Systems:** Inference (vLLM, TensorRT-LLM, Mooncake), RL Systems (Rollouts Optimization), Training (Megatron-LM).
- **Parallelism & Strategy:** Hybrid Parallelism, Disaggregated Serving, GPU Workload Balancing.
- **System Analysis & Optimization:** Theoretical Performance Modeling, Bottleneck Diagnosis, KV-Cache Management.
- **Languages & Hardware:** C++, Python, Bash, NVIDIA H100/H200, AWS Trainium.

## RESEARCH & PUBLICATION

Efficient long-context language model training by core attention disaggregation **[Accepted, MLSys26]**

• Designed and developed **Planner algorithm** to **balance attention workload** while minimizing the communication dispatch overhead, enabling precise FLOPS measurement via CP head-tail sharding. Integrated planner into 4D parallel for pretraining.

Seer: Online Context Learning for Fast Synchronous LLM Reinforcement Learning **[Submitted, OSDI26]**

Kimi Linear, Kimi K2.5: Visual Agentic Intelligence [Technical Report, 2026] (Contributor: Infrastructure Optimization)

## WORK EXPERIENCE

**Moonshot AI (Kimi)** | *vLLM, Mooncake, RL System* Remote | Jul 2025 – Present

• **RL System Optimization:** Led the integration of **Mooncake (Kimi's disaggregated inference engine)** into the RL framework, unlocking independent parallelism strategies for prefill and decode. Validated and deployed a hybrid configuration (**CP+TP** for prefill to enable Prefix Caching, **DP+EP** for decode to maximize throughput), overcoming legacy concurrency limits and accelerating FP8 Agentic rollouts by **1.5x (vs. monolithic vLLM CP+TP baseline)**.

• **Workload-Aware Performance Tuning:** Derived optimal **Prefill-to-Decode node ratios** (e.g., 3:5 for FP8) via theoretical modeling, successfully balancing **prefill throughput (waiting queue)** against **decode KV-Cache memory**. Tuned **EP degrees** and **A2A chunk sizes** to maximize **KV-cache capacity** while minimizing **dispatch overhead**.

• **Scheduler & Runtime Optimization:** Redesigned scheduler logic to budget based on **computational tokens** rather than total tokens. Resolved **scheduling stalls** caused by undersized memory guardrails during high-reuse scenarios**.** Coupled with **rollout-level concurrency control**, these optimizations boosted prefill throughput by **1.7x** with a **90%** KV-cache hit rate.

**AWS, Annapurna Labs** | *Applied Scientist Intern* | *Multimodal inference* GitHub Feb 2025 – May 2025

• **Accelerated Flux Image Generation:** Co-designed and implemented **Context Parallelism** (Sequence dimension splitting with All-Gather) on AWS Trainium 2. Reduced inference latency by **40% (7.9s → 4.8s)** via hybrid parallelism (TP8 + CP2), successfully **deploying the solution to production**.

• **Model Migration:** Led the migration of CLIP, T5, and VAE to ModelBuilder; resolved critical VAE accuracy regressions and backbone bottlenecks, establishing reusable **abstractions** for future onboarding.

**Nvidia, TensorRT-LLM** | *Computer Architecture Intern* | *LLM Inference* Sep 2024 – Jan 2025

• **Developed a comprehensive benchmarking framework** for multi-turn conversations (simulating Agentic RL) using **ShareGPT**. Extended **SGLang** DSL to support **TensorRT-LLM and vLLM**, enabling unified cross-framework evaluation.

• **Investigated** performance degradation in multi-turn scenarios on H200 by **profiling** TPS, TTFT and TPOT under varying concurrency. Revealed that excessive concurrency triggers **KV-cache eviction**, underscoring the need for concurrency control.

• Evaluated **LFU vs. LRU** KV-Cache eviction policies and identified that LRU outperforms LFU due to **strong temporal locality** in eviction patterns, securing optimal cache efficiency.

**Tencent, Cloud Architecture** | *Machine Learning Engineer Intern* | *LLM Inference* May 2024 – Aug 2024

• Built a robust **Beam Search** decoding feature, upgrading the framework from simple Greedy/Random sampling to supporting 2K candidate sequence maintenance.

• Implemented **Copy-On-Write** logic for beam search to trigger KV-Cache physical block duplication only during token forking, minimizing memory footprint for branching sequences.

• **Optimized Inference Latency:** Diagnosed CPU-fallback bottlenecks in sampling via profiling and implemented **Batched Prefill scheduling**, reducing prefill latency by **20.4%** (4.5s → 3.5s).