Physical AI Deployment & Inference Infrastructure Engineer

Alps Lab | Systems Engineer | Physical AI Deployment and Inference Infrastructure

Hiring in China, Singapore, North America and Europe · Remote

About the team

Alps Lab is building recursively self-improving Cognizers with scientific judgment, advancing model architectures native to Physical AI and hardware-software co-design. We aim to help models perform more complex tasks on robots and other physical platforms, and improve continuously through experiments, evaluation and feedback from real-world operation.

The role

You will deploy Physical AI models onto robots and other physical platforms and own the full inference path from sensor input to action output. This calls for deep work on runtimes, scheduling, memory and kernels, as well as a Harness that proactively identifies bottlenecks, validates changes and continuously optimizes the system.

For this role, the Harness covers deployment orchestration, context and skill management, tool use, observability, diagnostics and benchmarking. You will improve these mechanisms together with the inference infrastructure so that models run faster and more reliably on real hardware while preserving task quality.

Responsibilities

- **Deliver real embodied deployment pipelines.** Connect VLMs/VLAs, World Models or WAMs to sensors, robot middleware and control systems. Handle preprocessing, asynchronous inference, action chunking and feedback. Deliver working, diagnosable systems on Jetson Orin, Jetson Thor or other target platforms.
- **Optimize end-to-end inference performance.** Identify bottlenecks using the appropriate stack, including C++, CUDA, TensorRT and ONNX Runtime. Optimize execution graphs, operator fusion, memory layouts, data movement, concurrency and quantization for the model and platform, while validating numerical accuracy and task quality.
- **Optimize serving paths for generative models.** For applicable autoregressive components, improve prefill, decode, KV cache, prefix caching, batching and scheduling. Understand how frameworks such as vLLM, SGLang and TensorRT-LLM work, along with the limits of their hardware support.
- **Build a recursively improving deployment Harness.** Use execution records and profiler results to let the system propose improvements to skills, workflows, caching, scheduling or implementation. Compare quality, latency and cost on independent benchmarks, incorporate effective changes, and let the new Harness drive the next round of infrastructure optimization.
- **Establish rigorous performance and cost benchmarks.** Track model, runtime and task performance together. Fix hardware, precision, context length, concurrency, input modalities and power modes; record warm-up, cache state and statistical distributions. Avoid treating a single throughput number as evidence of a real deployment benefit.
- **Deliver optimizations in maintainable systems.** Establish continuous benchmarking, numerical correctness checks, task regression tests, observability and version rollback. Analyze performance limits with the model team and provide measurements to inform next-generation model architectures and hardware co-design.

Qualifications & application

What you bring

- Hands-on experience deploying models to robots, embedded platforms or other performance-sensitive systems, with ownership of the critical path from model export to execution on real hardware.
- Proficiency in modern C++, Python, Linux and concurrent programming. An understanding of memory management, CPU/GPU data flows and on-device resource constraints.
- Experience developing or optimizing CUDA kernels. You can use tools such as Nsight Systems/Compute to identify compute, memory-access, synchronization and kernel-launch overhead, and understand operator correctness and low-precision error.
- Familiarity with at least one production inference framework, including the ability to modify and debug it in depth. You understand prefill/decode, KV cache, scheduling and caching strategies for applicable models, and can assess their value on target platforms.
- Familiarity with inference and infrastructure benchmarks. You can report TTFT, TPOT/ITL, tokens/s, GPU memory usage, power, tail latency and cost per successful task. For non-token-based action models, you use appropriate metrics such as action latency, control frequency and task success rate.
- Familiarity with skill optimization and recursive Harness improvement, including Microsoft SkillOpt and Sakana AI/UC Berkeley RHI. You can turn feedback-driven optimization into an engineering process with baselines, versioning and regression validation.

Nice to have

- Experience with TensorRT plugins, Triton kernels, CUTLASS, model compilation or quantized deployment, with demonstrated end-to-end performance improvements and correctness validation.
- Familiarity with ROS 2, real-time robotic systems or asynchronous control, or experience with evaluation tools and methods such as GenAI-Perf and MLPerf Inference.
- Projects applying agents or automated optimization to performance diagnosis, experiment scheduling, code improvement and regression validation.

How we measure impact

Improvements in end-to-end latency, reliability, energy efficiency and cost per successful task, while preserving task success and numerical correctness. Optimizations should be reproducible across runs and validated under different inputs, loads or hardware conditions.

How to apply

Send your CV and 1-2 deployment or performance-optimization case studies. Describe the model and hardware, measurement conditions, before-and-after metrics, the code you changed, and how you verified that the speedup did not compromise numerical correctness or task quality.

We welcome candidates who demonstrate their capabilities through research, industry or open-source projects. You do not need to cover every nice-to-have.

contact@alps-lab.com

Related approaches: Microsoft SkillOpt · Recursive Harness Self-Improvement

Physical AI
部署与推理基础设施工程师

Alps Lab | Systems Engineer | Physical AI Deployment and Inference Infrastructure

招聘地区：中国、新加坡、北美、欧洲 工作方式：远程

关于团队

Alps Lab 正在构建具有科研判断能力、可递归自改进的认知体，推动 Physical AI 原生模型架构与软硬件协同设计。我们希望让模型在机器人等物理载体上完成更复杂的任务，并通过实验、评测和真实运行反馈持续改进。

关于岗位

你将把 Physical AI 模型部署到机器人等物理载体，并对从传感器输入到动作输出的整条推理链路负责。你需要深入运行时、调度、内存和 kernel，也需要构建能够主动发现瓶颈、验证修改并持续优化的 Harness。

本岗位的 Harness 包括部署编排、上下文与技能管理、工具调用、观测诊断及基准评测。你将让这些机制与推理基础设施共同改进，使模型在真实硬件上运行得更快、更稳，并保持任务质量。

你将负责

- **打通真实具身部署链路**。将 VLM／VLA、World Model 或 WAM 接入传感器、机器人中间件与控制系统，处理预处理、异步推理、动作分块和反馈；在 Jetson Orin、Jetson Thor 或其他目标平台上交付可运行、可诊断的系统。
- **优化端到端推理性能**。围绕 C++、CUDA、TensorRT、ONNX Runtime 等适用技术栈定位瓶颈；根据模型和平台优化执行图、算子融合、内存布局、数据搬运、并发与量化，并验证数值和任务质量。
- **优化生成式模型的服务路径**。针对适用的自回归组件优化 prefill、decode、KV cache、prefix caching、batching 与调度；理解 vLLM、SGLang、TensorRT-LLM 等框架的机制及硬件适配边界。
- **构建可递归改进的部署 Harness**。利用执行记录与 profiler 结果，让系统提出技能、工作流、缓存、调度或实现层的改进候选；在独立 benchmark 上比较质量、延迟与成本，纳入有效修改，让新的 Harness 继续驱动下一轮基础设施优化。
- **建立严谨的性能与成本基准**。同时跟踪模型、运行时和任务表现；固定硬件、精度、上下文长度、并发、输入模态和功耗模式，记录预热、缓存状态与统计分布，避免用单点吞吐替代真实部署收益。
- **把优化交付到可维护的系统**。建立持续 benchmark、数值正确性检查、任务回归、可观测性和版本回退；与模型团队共同分析性能限制，为下一代模型架构和硬件协同设计提供测量依据。

任职要求与申请方式

我们希望你具备

- 有将模型部署到机器人、嵌入式或其他性能敏感系统的 hands-on 经验，能够对从模型导出到真机运行的关键链路负责。
- 熟练掌握现代 C++、Python、Linux 与并发编程，理解内存管理、CPU／GPU 数据流及端侧资源约束。
- 有 CUDA kernel 开发或优化经验，能使用 Nsight Systems／Compute 等工具定位计算、访存、同步及 kernel launch 开销；理解算子正确性与低精度误差。
- 熟悉至少一种生产级推理框架，能深入修改和调试；理解适用模型的 prefill／decode、KV cache、调度与缓存策略，并能评估其在目标平台上的收益。
- 熟悉推理和基础设施 benchmark，能报告 TTFT、TPOT／ITL、tokens/s、显存、功耗、尾延迟及每次成功任务的成本；对非 token 动作模型使用动作延迟、控制频率和任务成功率等适用指标。
- 熟悉技能优化与递归 Harness 改进，包括 Microsoft SkillOpt 和 Sakana AI／UC Berkeley RHI；能把反馈驱动优化做成有基线、有版本、有回归验证的工程流程。

加分经历

- 有 TensorRT plugin、Triton kernel、CUTLASS、模型编译或量化部署经验，能展示端到端性能改善及正确性验证。
- 熟悉 ROS 2、实时机器人系统、异步控制，或使用过 GenAI-Perf、MLPerf Inference 等评测工具和方法。
- 有将 agent 或自动优化流程用于性能诊断、实验调度、代码改进和回归验证的项目经验。

我们如何衡量贡献

在保持任务成功率和数值正确性的条件下，改善端到端延迟、稳定性、能效与每次成功任务的成本。优化应能跨运行复现，并在不同输入、负载或硬件条件下经受验证。

申请方式

请提供简历，以及 1-2 个部署或性能优化案例。请说明模型与硬件、测量条件、优化前后指标、你修改的代码，以及如何确认提速没有损害数值正确性或任务质量。

我们欢迎通过研究、产业或开源项目证明能力的候选人，不要求你覆盖所有加分项。

contact@alps-lab.com

相关方法：Microsoft SkillOpt · Recursive Harness Self-Improvement