Efficient on-device LLM inference

Overview of 3 major directions: **Model/context efficiency**–reduce work and storage; **Inference execution**–execute work efficiently; **Resource management**–manage capacity and budgets.

Model/context efficiency	Inference execution	Resource management
Model design, quantization, compression	Kernels and compilers	Memory and storage offloading
Sparsity and conditional compute	**Heterogeneous execution** (Our focus)	Serving and resource sharing
Attention and KV efficiency	Speculative decoding	Energy and thermal control

SOTA in heterogeneous execution

Prior work	Established approach	Limitation for our direction
llm.npu	Chunk sharing, outlier handling, and out-of-order overlap	CPU attention. Online ordering uses offline costs.
HeteroInfer	GEMM partitions and fast synchronization	GPU attention. Length-aware splits use offline costs.
HeteroMosaic	Offline tuning of GEMM splits, overlap, and asymmetric chunks	GPU attention. Deploys pre-tuned configurations.

Gap: Attention stays outside the NPU. Costs may differ from offline profile at runtime.

Our work: adaptive heterogeneous prefill

Goal: reduce one request's long-context prefill latency on mobile GPU/NPU by balancing the workload dynamically with runtime feedback.

1. **Share attention, beyond GEMM balancing.**
   Reuse fixed-shape NPU tiles so spare NPU capacity can help GPU attention.

2. **Adapt during prefill.**
   Use measured task costs to follow speed changes from heat or contention.

Shorten the attention critical path and keep the split useful as conditions change.