ML Lab Research Proposal

Sushaan Kandukoori, Aarit Atreja, Avaneesh Parvathareddy, Devom Brahmbhatt

June 26, 2026

1. Problem Statement

Merging several LoRA adapters trained on the same base model usually means validating every candidate combination on held-out data, which gets expensive fast once a library has more than a handful of adapters. We built a screen that answers the cheaper question first: which subsets are even worth validating? It uses convex geometry to score each subset and returns a certificate, so the answer is checkable rather than a black-box guess. The work left is to confirm, by direct measurement instead of extrapolation, that the screen's scaling and its practical payoff hold up on real libraries.

2. Proposed Approach

Each task gets a quadratic surrogate of its retained loss over a shared space of merge coefficients, and we score a subset by its worst-case deficit, which is a convex minimax. That score carries a primal-dual certificate from Sion's theorem and a Helly-type bound: only a few tasks (d+1 of them, dropping to r+1 when the task Hessians share a low effective rank) can decide whether a subset is feasible. We picked convex geometry over a learned predictor for two reasons. It needs no per-library training data, so it works on a brand-new library where a gradient-boosted baseline has nothing to learn from, and it returns a certificate you can check instead of a bare number. The theory and most of the experiments are already done and written up. This grant funds the GPU runs that turn three claims from "extrapolated" into "measured."

3. Dataset

We use public LoRA adapters from Hugging Face that sit on shared base models, restricted to Apache-2.0 and MIT-compatible licenses, with no redistribution of weights. The main panel is a 50-adapter Qwen2.5-0.5B library with 4,000 merge events already evaluated, plus five smaller panels on TinyLlama, Mistral-7B, Phi-2, and two Pythia sizes. For the downstream experiment we assemble a mixed-task library from adapters we already have configs for (math, SQL, physics, Docker help, and a few others) and test on each task's held-out set, with a slice of GSM8K for the math axis. Preprocessing is light. We fit the quadratic surrogates from a small set of calibration prompts, project any indefinite Hessians onto the PSD cone, and solve the certificate.

4. Evaluation Metrics

We judge the screen the way the paper already does and add the metric specific to each new run.

- Screening quality: AUROC and AUPRC against held-out feasibility, per merge operator, with clustered and leave-one-adapter-out confidence intervals.
- Compression: the measured effective rank r and the realized core-table reduction at m=50, compared against the roughly 108× figure we currently extrapolate.
- Projection ablation: AUROC under the projected score, the un-projected jet, and their midpoint, on the events where projection actually fires.
- Downstream retention: held-out task accuracy, including GSM8K accuracy, for merges the screen picks versus merges a cosine or random baseline picks at the same size.
- Cost: GPU-hours saved at a fixed recall and precision operating point.

5. Compute Requirements

The base models are small. The headline work runs on Qwen2.5-0.5B in 4-bit, under 2 GB, so memory is never the limit and throughput is what matters. Anchored to the paper's own cost model of about 30 seconds per merged-model evaluation on an MI300X, the AMD estimates are:

- m=50 compression: 3 to 12 MI300X-hours, closer to 1 to 3 hours of wall-clock once we batch the evaluations and use the fact that merges are linear in the coefficients.
- Projection ablation: 10 to 33 MI300X-hours.
- Downstream retention: 3 to 8 MI300X-hours.

That comes to roughly 20 to 50 GPU-hours, which fits inside a single 8-GPU MI300X node over a couple of days. The stack is ROCm with PyTorch and PEFT, plus our own screening code. Since the model is this small, an MI210, an MI250, or even a consumer Radeon would run all of it; the bigger cards only buy back wall-clock.

6. Expected Deliverables

By the end of two weeks we expect:

- A revised paper, ready for TMLR, with the three extrapolated claims replaced by measured ones.
- A first downstream result showing that screen-picked merges keep their tasks better than baseline-picked merges of the same size, math included.
- Released code, the per-query certificate tables, and the manifests, so the numbers reproduce.

If those land early, the stretch goal is a small demo that takes a fresh adapter library and returns a ranked, certified shortlist of merges worth validating.

7. Risk & Mitigation

The thing most likely to disappoint is the downstream result, because a merge the screen greenlights could still lose accuracy on a real benchmark. We have two guards. We frame the experiment as retention under selection, which is the job the screen is built for and which its current AUROC of 0.76 to 0.89 already predicts it can do, rather than as a claim that merging adds new capability (that version was tested before and came back negative, for good reason). We also check the sign on a small library before committing the full budget, so a flat result costs a few hours instead of the grant.

The second risk is that the effective rank grows at m=50 and the compression looks weaker than the extrapolation. If that happens it is still a real measurement worth reporting, since it marks where the amortization actually helps, and the screening and certificate results hold up on their own.

The last risk is setup friction with ROCm or GPU access. The model is small enough to run on almost any AMD card, so the fallback is slower hardware, which costs wall-clock and nothing else.