ATLAS: Adaptive Task-aware Federated Learning with LoRA-based Heterogeneous Splitting

Supervisor Update — MIRA-aligned pipeline, fixes, and latest results

Advanced Master's Project

February 4, 2026

Agenda

1. What changed since midterm
2. Literature-grounded improvements (MIRA / HSplitLoRA alignment)
3. Engineering issues encountered & how we fixed them
4. Latest end-to-end results (Quick ATLAS run)
5. Next experiments (publication-quality evaluation plan)

What changed since January 7

Goal: move from toy runs (few clients/rounds) to a publishable, MIRA-faithful pipeline.

Major updates delivered:

- Real training on HF models + GLUE tasks (no synthetic curves)
- 9-client multi-task setup (3 tasks $\times$ 3 clients) with device heterogeneity
- Task-pure clustering from gradient fingerprints (privacy-preserving)
- Importance-aware per-layer LoRA ranks under memory budgets
- MIRA RBF adjacency + Laplacian personalization with block-diagonal graph

Current state: end-to-end ATLAS runs in ˜14 min for 3 rounds on a T4; ready for longer sweeps.

Quick run configuration (latest)

- Model: `distilbert-base-uncased`
- Tasks: `sst2`, `mrpc`, `cola`
- Clients: 9 total, `clients_per_task=3`
- Device types: `[2GB CPU, 4GB tablet, 8GB laptop, 16GB GPU]`
- Rounds: $T = 3$, local epochs $R = 2$, batch size 16
- Fingerprinting: 64 batches, PCA target 64D (uses 9 comps with 9 clients)
- Graph: `mira_rbf`, $\eta = 0.1$, block-diagonal, ensure connectivity

Repro command: `python experiments/atlas_integrated.py --quick --num-rounds 3`

Phase 1: Literature-grounded fingerprinting & clustering

Motivation (MIRA-style): cluster clients without seeing data, using task-informative gradients.

Implemented improvements:

- Extract gradients from last transformer layers + classifier (more task-specific)
- Increase fingerprint samples to reduce noise (64 batches)
- Per-layer L2 normalization to avoid domination by a single layer
- Multi-metric k-selection (Silhouette / Davies-Bouldin / Calinski-Harabasz)
- Singleton penalty to avoid fragmented clusters (prefer 1 cluster per task)

Phase 1: Latest clustering result (from quick run)

PCA: 9 samples, 14.8M features, 9 components (top-3 explain 0.472).

k-search (singleton penalty active):

k	Combined	Silhouette	DB	Singletons
2	0.363	0.051	1.994	0
3	**0.382**	**0.071**	**1.639**	**0**
4	0.244	0.052	1.300	1
5	0.106	0.040	1.061	2

Selected: $k = 3$ with **task-pure clusters** (purity = 1.0)

- Cluster 0: MRPC clients [3,4,5]
- Cluster 1: CoLA clients [6,7,8]
- Cluster 2: SST-2 clients [0,1,2]

Phase 2: What went wrong & the fix

Problem we hit: we computed per-layer importance scores correctly, but ranks stayed uniform (e.g., [8,8,8,8,8,8]).

Root cause: incremental greedy upgrades often let *every* layer reach the same ceiling rank if the memory check is permissive.

Fix implemented: budget-proportional allocation

- Find max *uniform* rank that fits memory budget (baseline)
- Convert that to a *total rank budget*
- Allocate per-layer ranks proportional to importance, then round to candidates
- Validate memory; if needed, downgrade least-important layers

Reference: HSplitLoRA constraint $\sum_{\ell} 2dr_{\ell}b \leq C_{mem}$ with $C_{mem} = M_{device}(1 - \alpha_{base} - \alpha_{act} - \alpha_{opt})$.

Phase 2: Budget-proportional allocator (pseudo-code)

Algorithm 1 Importance-aware rank allocation (budget-proportional)

Compute adapter budget C_{mem} from device memory and $(\alpha_{base}, \alpha_{act}, \alpha_{opt})$
Find best uniform rank r^* s.t. $n_{alloc} \cdot M(r^*) \leq C_{mem}$
Total rank budget: $B \leftarrow n_{alloc} \cdot r^*$
for each layer ℓ **do**
    $\tilde{r}_\ell \leftarrow \text{importance}_\ell \cdot B$
    $r_\ell \leftarrow \text{round\_to\_candidates}(\tilde{r}_\ell)$
end for
if $\sum_\ell M(r_\ell) > C_{mem}$ **then**
    Downgrade least-important layers until feasible
end if
return $\{r_\ell\}$

Phase 2: Latest per-layer ranks by device (examples)

Device	Example ranks (6 LoRA layers)	Adapter mem	Notes
2GB CPU	[4, 8, 8, 8, 4, 4]	0.21MB	lowest comm cost
4GB tablet	[8, 16, 16, 16, 4, 4]	0.38MB	moderate capacity
8GB laptop	[16, 32, 32, 32, 4, 4]	0.70MB	higher ranks mid/late
16GB GPU	[32, 64, 64, 64, 4, 4]	1.36MB	highest capacity

Observed importance pattern (client 0): layer_3 $>$ layer_2 $>$ layer_1 $>$ layer_0 $\gg$ layer_4 $>$ layer_5.

Communication scaling with heterogeneity (per round)

Observation: communication cost scales with rank and device capacity.

Device type	Upload (bytes)	Download (bytes)
2GB CPU	5,621,776	1,769,472
4GB tablet	6,506,512	3,538,944
8GB laptop	8,275,984	7,077,888
16GB GPU	11,814,928	7,077,888

Interpretation: split + LoRA keeps costs far below full-model FL; larger devices contribute more.

Phase 4: MIRA RBF adjacency + Laplacian personalization

MIRA adjacency (implemented):

$$a_{k\ell} = \exp\left(-\alpha \|f_k - f_\ell\|^2\right), \quad \sum_{\ell \in N_k} a_{k\ell} = 1$$

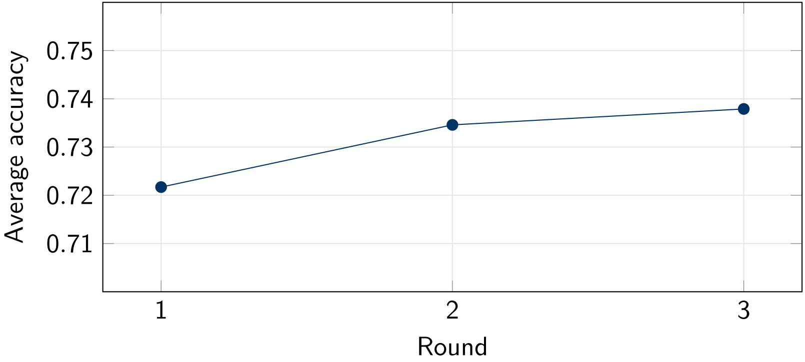
Personalized update (per client):

$$W_k^{(t+1)} = W_k^{(t,R)} - \eta \sum_{\ell \in N_k} a_{k\ell} \left(W_k^{(t,R)} - W_\ell^{(t,R)} \right)$$

Latest run:

- Block-diagonal graph (no cross-task mixing)
- Full intra-cluster connectivity with $k = 3$ and clusters of size 3
- **18 directed adjacency weights** computed (6 per cluster)

End-to-end results: accuracy improves each round



Runtime: ~203 sec/round on T4; total ~13.9 minutes for 3 rounds.

Final accuracy snapshot (Quick ATLAS run)

Final per-client accuracy (round 3):

- SST-2 (clients 0–2): 0.826, 0.828, 0.827 (avg ˜0.827)
- MRPC (clients 3–5): 0.711, 0.689, 0.684 (avg ˜0.695)
- CoLA (clients 6–8): 0.692, 0.694, 0.691 (avg ˜0.693)

Overall average accuracy: 0.738

Note: MRPC/CoLA are harder tasks; expect larger gains with $T \geq 20$ rounds and η sweep.

Engineering issues we faced (and resolved)

1. **Toy setup / weak clustering:** too few clients, too few fingerprint batches
   - Fix: 9 clients (3 tasks $\times$ 3), 64 fingerprint batches
2. **Cluster fragmentation:** k-search picked k=5 with singleton clusters
   - Fix: singleton penalty in clustering score $\Rightarrow$ k=3 selected
3. **Uniform ranks despite importance:** allocator upgraded all layers equally
   - Fix: budget-proportional rank allocation (per-layer heterogeneity)
4. **MIRA graph connectivity:** needed consistent neighborhoods
   - Fix: full intra-cluster connectivity; ensure connectivity enabled
5. **Real-world debugging:** configuration drift + JSON formatting bugs during iteration
   - Fix: config logging + strict result saving; quick mode stabilized

Next experiments (Feb 2026 evaluation plan)

Goal: quantify benefit of Laplacian personalization and hetero ranks.

- **Longer runs:** $T = 20$ and optionally $T = 60$ (MIRA shows clearer gains after ˜20)
- η **(lambda) sweep:** $\eta \in \{0.0, 0.01, 0.1, 0.5, 1.0\}$
- **Ablations:**
  - (i) no Laplacian ($\eta = 0$), (ii) FedAvg-in-cluster baseline, (iii) full ATLAS
- **Robustness:** 3 random seeds, report mean $\pm$ std and worst-client accuracy
- **Rank quantization study:** denser rank candidates to reduce ties (e.g., 4/6/8/12/16/24/32/48/64)
- **Metrics:** track per-task accuracy, F1 (MRPC), and fairness (worst client)

Discussion points for supervisors

- Target evaluation: more tasks/clients vs deeper tuning on 3 GLUE tasks?
- Preferred baselines: FedAvg + LoRA, per-task FedAvg, or local-only?
- Desired reporting: comm cost (bytes/round), wall-clock time, and accuracy tradeoffs
- What constitutes "publishable" scale for this project (clients/rounds/seeds)?

Thank You

Questions & Feedback