Unified-LoRA — Technical Report & Mathematical Analysis

1. Introduction

LoRA (Low-Rank Adaptation) is widely used for efficient fine-tuning of large models. Standard LoRA training relies on *fixed learning rate schedules* and a *static update rule* that does not react to stress, distribution shocks, or changing task difficulty.

Unified-LoRA introduces a **dynamic controller** that adapts the LoRA training regime in real time using a smooth stress signal ϕ and a 3-state finite state machine (FSM). The controller selects among:

- **Single LoRA** – aggressive adaptation
- **Multi LoRA** – moderate adaptation
- **Mirror LoRA** – conservative, stable mode

The transition between modes is automatically driven by a signal computed from the model's loss and environmental stress.

Goal:

> Make LoRA training **adaptive**, **stable**, and **shock-resilient**, without modifying the model architecture.

---

2. Mathematical Foundations

2.1 Stress observation

Let:

- (L_t) = model loss at step (t)
- (E_t) = external/environmental stress (optional)
- (_t) = smoothed combined stress

We define a normalized loss:

$$D_t = \frac{L_t}{1 + L_t}$$

and a combined observation:

$$X_t = \alpha_E E_t + (1 - \alpha_E) D_t$$

The smoothed stress obeys the exponential filter:

$$\hat{E}_{t+1} = \beta \hat{E}_t + (1 - \beta) X_t$$

with typical values:

- (= 0.9)
- (_E = 0.5)

This gives the system **memory**, preventing oscillations.

2.2 Stress-to-mode signal ϕ

We define the *stress scalar* (_t) as:

$$\phi_{t+1} = (1 - \alpha)\phi_t + \alpha \frac{\hat{E}_t}{1 + \hat{E}_t}$$

where:

- ( ) controls responsiveness
- the denominator guarantees ((0,1))

2.3 Mode Switching (FSM)

We define:

- (_0 = 0.3) – threshold to leave aggressive mode
- (_1 = 0.7) – threshold to enter conservative mode
- (N_s) – minimum cooldown before another state change

The FSM is:

$$M_{t+1} = \{0, \quad \phi_t < \theta_0 \; 1, \quad \theta_0 \le \phi_t < \theta_1 \; 2, \quad \phi_t \ge \theta_1$$

Mode meanings:

Mode	Name	Behavior
0	Single	Highest LR, fast adaptation
1	Multi	Moderate LR
2	Mirror	Lowest LR, stable

Cooldown constraint:

$$t - t_{\text{last\_switch}} \ge N_s$$

2.4 LR Scheduling Coupled to Mode

Learning rate is selected as:

$$\eta_t = \{\eta_0, \quad M_t = 0\ \eta_1, \quad M_t = 1\ \eta_2, \quad M_t = 2$$

With:

- (_0 = 2^{-3})
- (_1 = 5^{-4})
- (_2 = 1^{-4})

This couples *system stress* to *training aggressiveness*.

---

3. Dynamics & Stability Analysis

3.1 Low-Stress Regime

If shocks are absent, (_t X –0.5), and

$$\phi_t \to \phi^* < \theta_0$$

System converges to:

- **M = 0**
- aggressive learning rate (_0)
- rapid convergence of LoRA weights

This corresponds to *efficient training*.

3.2 Shock Regime

If loss spikes:

$$L_t \gg 1 \Rightarrow D_t \approx 1$$

$$X_t \to 0.5E_t + 0.5$$

$$\phi_t \to \phi^* > \theta_1$$

Then the FSM transitions to:

Single → Multi → Mirror

Thus the system:

- lowers LR,
- increases stability,
- slows parameter drift,
- prevents catastrophic forgetting.

3.3 Recovery Regime

Because of smoothing:

$$\hat{E}_t \quad \text{decays slowly}$$

Thus:

- φ stays high temporarily,
- system remains in **conservative mode** even after shock ends,
- preventing overshoot.

This hysteresis is essential for **robustness**.

---

4. Implementation

The controller is implemented in ~60 lines:

- ϕ update
- stress smoothing
- FSM with cooldown
- LR mapping

Works with any LoRA backend. Verified integration with Tinker's LoRA API via:

- `create_lora_training_client`
- `forward_backward`
- `optim_step`

The controller is *stateless across steps except for* φ, so it can be plugged in anywhere.

---

5. Experimental Evaluation on Tinker

5.1 Setup

- Model: `meta-llama/Llama-3.2-1B`
- API: Tinker LoRA fine-tuning
- Task: Pig-Latin mapping
- Shocks:
  - 200–300
  - 500–600
- Dataset:

- clean examples
- corrupted “shock” examples

Two runs:

1. **Unified-LoRA (adaptive controller)**
2. **Baseline (fixed LR = 5e-4)**

5.2 Unified-LoRA Results

Key behavior:

- detects shock
- increases φ
- transitions Single → Multi → Mirror
- reduces LR accordingly
- stabilizes loss extremely fast
- no post-shock overshoot

Example:

Step	Shock	Loss	Mode	LR
200	Yes	18.42	M=0 → M=1	↓
225	Yes	2.56	Multi	5e-4
250	Yes	0.0015	Multi	5e-4
275	Yes	0.0010	Mirror	1e-4
350	No	**0.0004**	Single	2e-3

Recovery complete by step 350.

5.3 Baseline Results (LR fixed)

Step	Shock	Loss
200	Yes	9.28
225	Yes	1.89
250	Yes	3.43 □ rebound
300	No	**13.09** □ huge overshoot
350	No	3.70

Observations:

- unstable under shock
- strong oscillations
- catastrophic overshoot *after shock ends*
- slow recovery (loss still >1 at step 700)

6. Conclusions

Unified-LoRA introduces **adaptive, stress-driven control** over LoRA training. Key advantages demonstrated:

- **automatic reaction to shocks** via stress signal ϕ
- **dynamic LR modulation**
- **stability through hysteresis**
- **seamless integration** with existing LoRA APIs
- **faster loss stabilization** (vs fixed-LR baseline)
- **avoidance of post-shock overshoot**

This behavior is desirable in real-world online fine-tuning scenarios where:

- data shifts,
- noise bursts,
- and unstable gradients are common.

Unified-LoRA behaves as a **meta-controller** on top of LoRA, improving robustness without architectural changes.