Developers Training Handbook for ByteDance Model Training Project

Module 1: Project Mission, Workflow & Lifecycle

Introduction

This module serves as the foundational onboarding guide for all stakeholders, including Team Leads, Prompt Generators, and Developers. It defines the technical mission of the project and the rigorous lifecycle every data point must undergo to ensure the successful evolution of the ByteDance Seed model.

1. Project Context

The objective of this initiative is the targeted advancement of the ByteDance Seed model. While the model currently demonstrates high-level proficiency in general coding tasks, it exhibits specific logical "lags" when benchmarked against global State-of-the-Art (SOTA) models such as the latest versions of ChatGPT and Claude.

Our mission is to identify the precise boundaries of these capabilities and provide high-density, synthetic data to bridge that gap. We are not merely building a collection of code; we are engineering a specialized curriculum designed to push the model toward the absolute edge of its logical capacity.

2. Statistical Failure Verification

To ensure maximum training efficiency, we do not produce data for problems the model can already solve. We utilize a strict protocol to calibrate the difficulty of every task:

- Every proposed coding challenge is first executed against the ByteDance Seed model for five independent iterations. A prompt is only valid if the model fails to produce a correct solution in at least two out of the five attempts. This ensures we target consistent reasoning deficits rather than stochastic errors.
- Simultaneously, the prompt must be verified as solvable by a SOTA model. If external models also fail, the prompt is deemed "Too Cold" (logically flawed or ambiguous) and is discarded.
- We focus exclusively on the complexity range where the Seed model fails but a superior intelligence succeeds.

3. The End-to-End Workflow & Lifecycle

The production of a single training data point follows a linear, rigorous pipeline. Stakeholders must ensure no noise or ambiguity is introduced at any stage.

Phase I: Prompt Engineering

Prompt generators generate highly specific, objective prompts. These are formal engineering requirements that define the problem space with absolute precision, including function signatures, time/space complexity constraints, and edge-case handling requirements.

Phase II: Verification & Calibration

As mentioned before, the prompt is tested against the Seed model (2/5 rule) and SOTA benchmarks. This phase identifies the Reasoning Pivot and the exact point where the model's logic breaks.

Phase III: Ground Truth Engineering

Developers implement the Ground Truth (GT). This is the definitive reference solution the model will emulate during Supervised Fine-Tuning (SFT). At this stage:

- The code must be simple, clear, and idiomatic.
- Solutions must produce identical results across all executions. Reliance on system time, unseeded randomness, or external networks is strictly prohibited.

Phase IV: Test Suite Construction

For every GT, a comprehensive test suite is developed. In Reinforcement Learning (RL), these tests act as the Reward Function. The model is rewarded if it passes and penalized if it fails. For Testing category prompts, developers must write Meta-Tests to verify the validity of the developed tests.

Phase V: Containerization & Integration

To ensure reproducibility across training clusters, every task is encapsulated in a Docker container. This freezes the tech stack and ensures Bit-Level Reproducibility. Files must be organized according to the standardized directory structure to allow for automated ingestion into the dataset.