TrainCheck

Training with Confidence: Catching Silent Errors in Deep Learning Training With Automated Proactive Checks

Yuxuan Jiang, Ziming Zhou, Boyu Xu, Beijie Liu, Runhui Xu, Ryan Huang

ARTIFACT EVALUATED usenix ASSOCIATION AVAILABLE

ARTIFACT EVALUATED usenix ASSOCIATION FUNCTIONAL

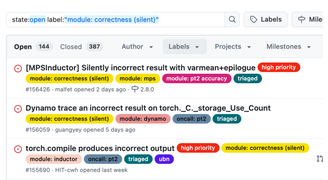
ARTIFACT EVALUATED usenix ASSOCIATION REPRODUCED

Preprint | Software | Reading List

Problem Statement

Silent training errors are **prevalent**, **hard to catch**, and **extremely costly**

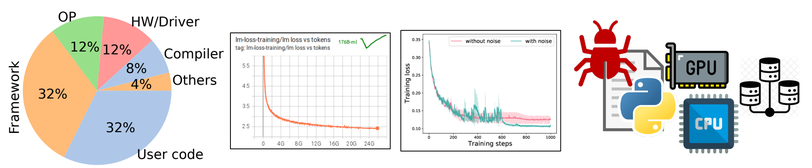
All major practitioners report silent training errors



PyTorch Meta Bloomberg 上海人工智能实验室 Shanghai Artificial Intelligence Laboratory

🤔 Can we detect silent training errors before it's too late?

Empirical Study Findings



1. Root causes are diverse and widespread
2. Hard to detect & severe impact
3. Many silent errors have **precise, actionable** root causes

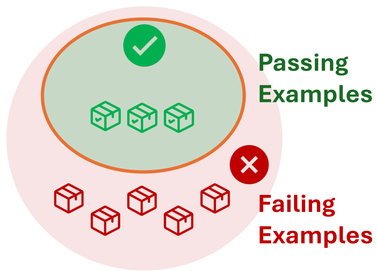
Core Insight: Non-determinism is an artifact of checking at too high levels

Inferring Transferable & Context Sensitive Semantics

`Inv: Relation(Desc1, Desc2), Precondition`

Relation	Description
`Consistent(Va,Vb)`	Va and Vb should have the same values, while the values may change
`EventContain(Ea, Eb)`	Eb must happen in the duration of Ea
`APISequence(Ia, Ib, ...)`	Ia, Ib, . . . must all occur and in the specified order
`APIArg(Ia, is_distinct)`	Ensures argument consistency or distinction in all calls to Ia
`APIOutput(Ia, bound_type)`	The output of Ia must meet certain attribute constraints

For each hypothesis, infer preconditions



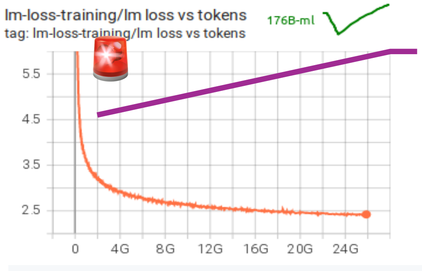
From the problem to the solution

- **Empirical Study:** Analyzed 88 real-world silent errors
- **TrainCheck:** Early Detection for Silent Training Errors
  - Learn **transferable** invariants from tutorial pipelines

Case Study – Bloom Parameter Divergence

BLOOM (176B) – **384** A100 GPU, **3.5** months

✅ *"Loss curve looks healthy"*



🚨 Weights diverged across tensor parallel groups

- Checkpoints became invalid

⚠️ *Could've wasted 3.5 months & 384 A100s*

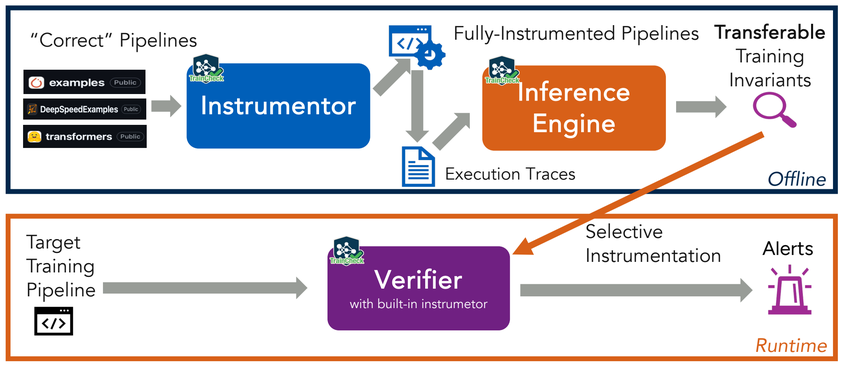
```python
@torch.no_grad()
def get_grads_for_norm(self, for_clipping=False):
    grads = []
    tensor_mp_rank = bwc_tensor_model_parallel_rank(mpu=self.mpu)
    for i, group in enumerate(self.bf16_groups):
        for j, lp in enumerate(group):
            if not for_clipping:
                if hasattr(lp, PIPE_REPLICATED) and lp.ds_pipe_replicated:
                    continue

            if not (tensor_mp_rank == 0 or is_model_parallel_parameter(lp)):
                continue # YUXUAN: as compared to the original code, this line is moved out by one indentati

            if not self.fp32_groups_has_gradients[i][j]:
                continue
```

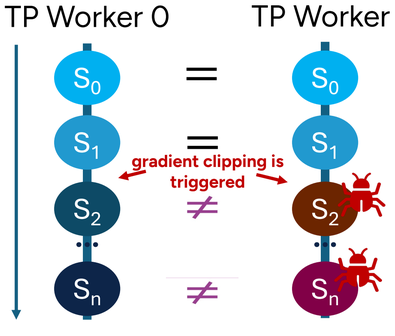
TrainCheck Design

✅ 1. Invariant (Semantic) Learning from Good Runs

✅ 2. Proactive Runtime Validation With Inferred Invariants

✅ 3. Systematically Covers Diverse Root Causes





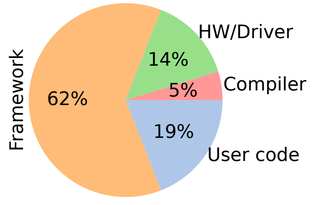
Training Invariant Examples

1. API Behavior Invariant
   `get_grad_for_norm` API contract

2. State Relationship:
   Parameters should be equal across workers

Evaluation

Failure Benchmark:
20 real-world silent training errors (14 newly collected)



✅ TrainCheck detects 18 out of 20 within 1 iteration

✅ TrainCheck pinpoints the exact root cause in 10 cases

✅ TrainCheck consistently shows < 2% FP rate with 5 representative input pipelines

✅ TrainCheck consistently detects 14 bugs with **5 randomly sampled inputs**

✅ Typical checking stage (selective with 100 invariants deployed) is **< 11%**