

Fresh-State Distillation for Tool-Error Recovery on a Consumer GPU

RecoveryBench v1 technical report - miniVERL v0.3.0 - Daoyuan Li

Primary finding: under eight equal continuation updates, strict fresh-state OPD underperformed frozen-student-state KD on both preregistered primary endpoints. Fresh-state supervision also required about 13.2x the continuation time.

Abstract

RecoveryBench isolates whether teacher supervision on states freshly visited by the current student improves recovery from structured SQLite tool errors, relative to distillation on a fixed state set collected from the cold-start student. We evaluate six equal-update arms and three-method selected-position and wall-time views over three prespecified seeds and 128 paired test tasks. A protocol-qualified Qwen3-1.7B teacher is selected on eval only and frozen at an immutable adapter revision. Fresh OPD does not improve the primary result: its strict-success mean is 10.9%, versus 23.2% for frozen KD, with a paired difference of -12.24 points. Recovery after error similarly falls by 13.79 points. The result is a scoped negative mechanism study, not a claim that OPD is universally ineffective or interchangeable with SFT.

Keywords: on-policy distillation, tool use, recovery, frozen states, consumer GPU

1. Question and hypotheses

The primary question is whether supervision on fresh states visited by the current student improves tool-error recovery over teacher scoring on states frozen from the shared cold-start student. H1 predicts a recovery advantage for strict fresh OPD under equal updates. H2 asks whether querying at most 50% of model-generated positions preserves most of any full-OPD gain. H3 requires quality to be interpreted with teacher queries, GPU time and peak VRAM.

RecoveryBench is not an alignment benchmark. SFT establishes basic capability and protocol competence; OPD is a teacher-student mechanism whose transferred behavior depends on the teacher. Ordinary task success is not a complete alignment objective.

2. Environment and controlled intervention

The new `sqlite_recovery` environment retains in-memory, read-only SQLite safety constraints and uses 12 deterministic structural templates split disjointly across train, eval and test. Tasks cover schema inspection, filtering, joins, aggregation and correction. One subset receives a deterministic one-time `SCHEMA_REFRESH_REQUIRED` error; another exposes natural SQL errors; a third has no intervention. Structured `error_code`, `retryable` and `intervention` fields make recovery metrics independent of fragile string matching.

All methods use the same 256-task training schedule and the same 128 test task IDs within each seed. Oracle intervention traces execute the failed query, inspect the schema, retry correctly and emit a final answer; SFT therefore trains a real recovery sequence rather than bypassing the error.

3. Methods

Method	State source	Supervision	Freshness
continued SFT	oracle	hard oracle tokens	fixed demonstration
oracle offline KD	oracle	teacher top-k + tail	fixed oracle
frozen-student KD	cold student	teacher top-k + tail	fixed once
strict fresh OPD	current student	teacher top-k + tail	each update
fresh OPD 50%	current student	teacher on selected positions	each update

Each seed starts from one shared 24-update SFT checkpoint. Frozen-student KD collects 64 cold-policy trajectories once at parameter version zero, scores them with the qualified teacher, and reuses the immutable dataset for all updates. Strict OPD recollects after every parameter update. Audit logs verify that rollout policy, current policy and current parameter versions match at all 75 fresh-OPD updates across the three result views.

4. Teacher qualification

The historical calculator protocol teacher was evaluated first on the 96-task RecoveryBench eval split and failed. Candidate A was trained by QLoRA on protocol-v2 oracle traces, reloaded on its preregistered NF4 base and became the first candidate to pass every gate. Candidate selection never used the test split.

Candidate	Strict	Recovery	Parse-valid	Tool success	Gate
calculator protocol teacher	25.0%	10.7%	95.8%	14.4%	fail
SQLite candidate A (NF4)	90.6%	81.2%	100.0%	87.1%	pass

Selected adapter: DaoyuanLi/mini-verl-qwen3-1.7b-sqlite-recovery-teacher at eb2747895ec32dab47c5b50c2d8aa9c0d9701e0d; weights SHA-256 5355f7007efb904d1b45a1aeb9b73b479b6f52025ab92502ab7895706155b2ba.

5. Budget matching

The primary view fixes eight continuation optimizer updates. The selected-token view stops at the first optimizer boundary at or beyond 6,224 selected positions and records overshoot. Every core method stopped after eight updates; overshoot was 0-646 positions.

The preregistered wall target is 50 continuation seconds. Fresh OPD crosses it in one indivisible step. SFT and frozen KD complete the eight-cycle ceiling before their internal continuation timer crosses 50 seconds, although complete train calls average 51.92 and 51.63 seconds because final checkpoint persistence is included there. The resulting artifact is a cycle-capped wall diagnostic, not exact equal-time evidence. It is preserved without a post-outcome rerun.

Teacher preparation is excluded from arm continuation time and reported separately: 2,310.75 seconds once; 462.15 seconds amortized over five students; 231.07 seconds over ten. An excluded 687.05-second diagnostic remains recorded.

6. Equal-update results

Method	Strict by seed (%)	Strict mean	Recovery	Train s	VRAM GiB
cold start	5.5, 5.5, 21.1	10.7%	13.6%	0.2	1.85
continued SFT	0.0, 5.5, 9.4	4.9%	1.8%	51.3	1.88
oracle offline KD	44.5, 0.0, 54.7	33.1%	31.9%	58.3	3.47
frozen-student KD	36.7, 9.4, 23.4	23.2%	22.8%	52.1	3.47
strict fresh OPD	26.6, 0.0, 6.2	10.9%	9.1%	686.8	3.47
fresh OPD 50%	21.9, 10.9, 49.2	27.3%	20.7%	720.8	3.47



Cyan is strict task success; violet is recovery after a structured tool error. Values are three-seed means. Seed order is 1234, 20260727, 20260801.

Paired task analysis

Endpoint	Fresh - frozen	95% paired bootstrap	Pairs
strict task success	-12.24 pp	[-15.89, -8.59]	384
recovery after error	-13.79 pp	[-20.69, -6.90]	116

H1 is not supported. Frozen-student KD has higher strict success and recovery than strict fresh OPD. Oracle-state offline KD has the highest mean. The 50% fresh arm reaches a 27.3% strict mean but ranges from 10.9% to 49.2%, so it does not establish a stable query-budget benefit.

7. Secondary budget views

Equal selected positions (target 6,224)

Method	Actual by seed	Strict	Recovery	Train s
continued SFT	6675, 6870, 6701	4.9%	1.8%	51.4
frozen-student KD	6467, 6753, 6658	23.2%	22.8%	51.3
strict fresh OPD	6224, 6822, 6290	10.9%	9.1%	683.3

Wall-time target diagnostic

Method	Steps by seed	Recorded stop kind	Strict	Train s
continued SFT	8, 8, 8	cycle cap (3/3)	4.9%	51.9
frozen-student KD	8, 8, 8	cycle cap (3/3)	23.2%	51.6
strict fresh OPD	1, 1, 1	wall target (3/3)	7.3%	136.7

Fresh OPD's one-step train calls average 136.65 seconds and reach 7.3% strict success. SFT and frozen KD remain eight-step outcomes because the configured cycle ceiling terminates them before the internal timer. No exact equal-time ranking is claimed.

8. Cost and failure analysis

At equal updates, strict OPD averages 686.80 continuation seconds versus 52.10 for frozen KD, about 13.2x. Peak reserved VRAM is similar for the KD/OPD methods at roughly 3.47 GiB. The position-budget arm queries 49.77% of model-generated positions but is not faster: compressed target selection reduces stored or projected positions, not teacher backbone forward cost.

Seed variation is large. Oracle KD ranges from 0.0% to 54.7% strict success, frozen KD from 9.4% to 36.7%, full fresh OPD from 0.0% to 26.6%, and the 50% arm from 10.9% to 49.2%. The data therefore support a scoped negative comparison, not a universal ordering of algorithms.

9. Limitations

The study uses one Qwen3 student/teacher pair, one read-only SQLite recovery environment, one RTX 4080 and three seeds. It evaluates mechanism-level task success and recovery, not safety, preference, over-refusal or general utility. The teacher is qualified on 96 eval tasks, not proven universally competent. Final-test generation is deterministic, but training remains sensitive to seed and finite task schedules.

The wall-time secondary view is cycle-capped for six runs and must not be read as an exact equal-wall experiment. Task-paired bootstrap intervals condition on this fixed task set and do not replace replication across models and environments. The 50% selector changes queried positions but not teacher forward count. No claim of statistical significance is made solely from three seeds.

10. Reproducibility and artifact integrity

Preregistration commit: 7087b3a333463b88a62ffed73daee2c85d039145. Revision-1.3 digest: 9c4c2ec19a56cebb2b2c1c0f3c7e504a9285467c99ae1590488251fbf2ff3934. Final execution commit recorded by the results: dba595f55ac634c7e0735db696a052126994bb26. The selected teacher adapter revision and weight digest are recorded in every KD/OPD arm and every frozen dataset manifest.

Artifact	SHA-256
recoverybench-v1-equal-updates.json	6ce2e6837e12b99ebc4fad6d27ce3e69c92e295ff3b9b60e0f68c2d308022384
recoverybench-v1-equal-selected-tokens.json	fe4c9afc799724dfe7a32e631676a1e5177c44559a7374d2ea31da135354f137
recoverybench-v1-equal-wall-time.json	425b0fa568f37b09e61af731d3da5009bd3833bddde6efaf2c66e9dba8355cbe
recoverybench-v1-analysis.json	8a6891f74aed80f07ec00d5ea1909895c579346e1abbb1d5d95a354bb46c6b81
recoverybench-v1-task-results.jsonl	aff96bffc6da27240a852410ac041bd4d95badf34cad030e6f437be1491a55ad
legacy calculator result	53fc1d4d5b7adee09618d77ad62d4086ba56b78569832d6fc7c3bcd5c2695bbc

Audit coverage: 36 task artifacts, 4,608 task trajectories and 101,787,618 raw bytes; every embedded SHA-256 and byte count matched. Task IDs are paired across all methods within seed. Three cold checkpoints and three frozen-student dataset digests are shared across budget views. The legacy calculator JSON remains byte-identical.

11. Conclusion

RecoveryBench finds no fresh-state advantage in this setting. Frozen-student KD is better and far cheaper than strict fresh OPD under the preregistered primary comparison. The result narrows miniVERL's product claim: it is an independent one-GPU companion for prototyping, diagnosing and validating online post-training workflows, not a claim that OPD replaces SFT or that local execution is equivalent to a distributed verl runtime.