A New Paradigm in Tuning Learned Indexes - A Reinforcement Learning Enhanced Approach

Taiyi Wang[1], Liang Liang[2], Guang Yang[3], Thomas Heinis[3],

and Eiko Yoneki[1*]

[1] University of Cambridge

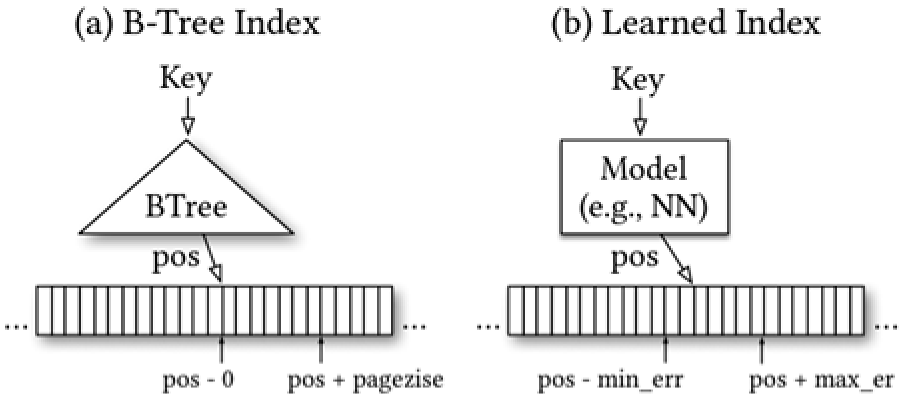
[2] EPFL

[3] Imperial College London

Introduction: Learned Index Structure (LIS) is trendy

Learned Index: Model-Driven Indexing

- Models capture data distribution using empirical cumulative distribution function
- Approximates key locations to reduce search operations
- Performance depends heavily on model accuracy for data patterns



	ART	B-tree	BS	FAST	IS	RBS	RMI	RS	TIP
amzn32	n/a	529	773	**244**	4604	325	264	275	731
face32	**187**	524	771	229	1285	312	274	386	964
logn32	n/a	522	765	294	n/a	471	**97.0**	105	744
norm32	191	522	771	229	10257	355	**71.7**	**70.9**	884
uden32	102	521	771	228	**39.8**	333	54.2	64.2	176
uspr32	n/a	524	771	230	469	301	**153**	200	400
size overhead	47%	16%	0%	123%	0%	< 1%	3%	< 1%	0%
amzn64	n/a	601	804	n/a	4736	387	266	**288**	759
face64	391	592	784	n/a	1893	**337**	**334**	461	1232
logn64	309	597	784	n/a	n/a	753	179	**120**	454
norm64	266	592	785	n/a	10510	405	**71.5**	**70.5**	862
osmc64	n/a	599	785	n/a	95076	492	**402**	437	7186
uden64	112	592	784	n/a	**43.4**	344	54.3	53.9	193
uspr64	287	591	785	n/a	449	313	**169**	214	428
wiki64	n/a	608	802	n/a	7846	364	**222**	**218**	1019
size overhead	25%	16%	0%	n/a	0%	< 1%	3%	< 1%	0%

Different LIS performance under different data distributions

Introduction: Interest Findings

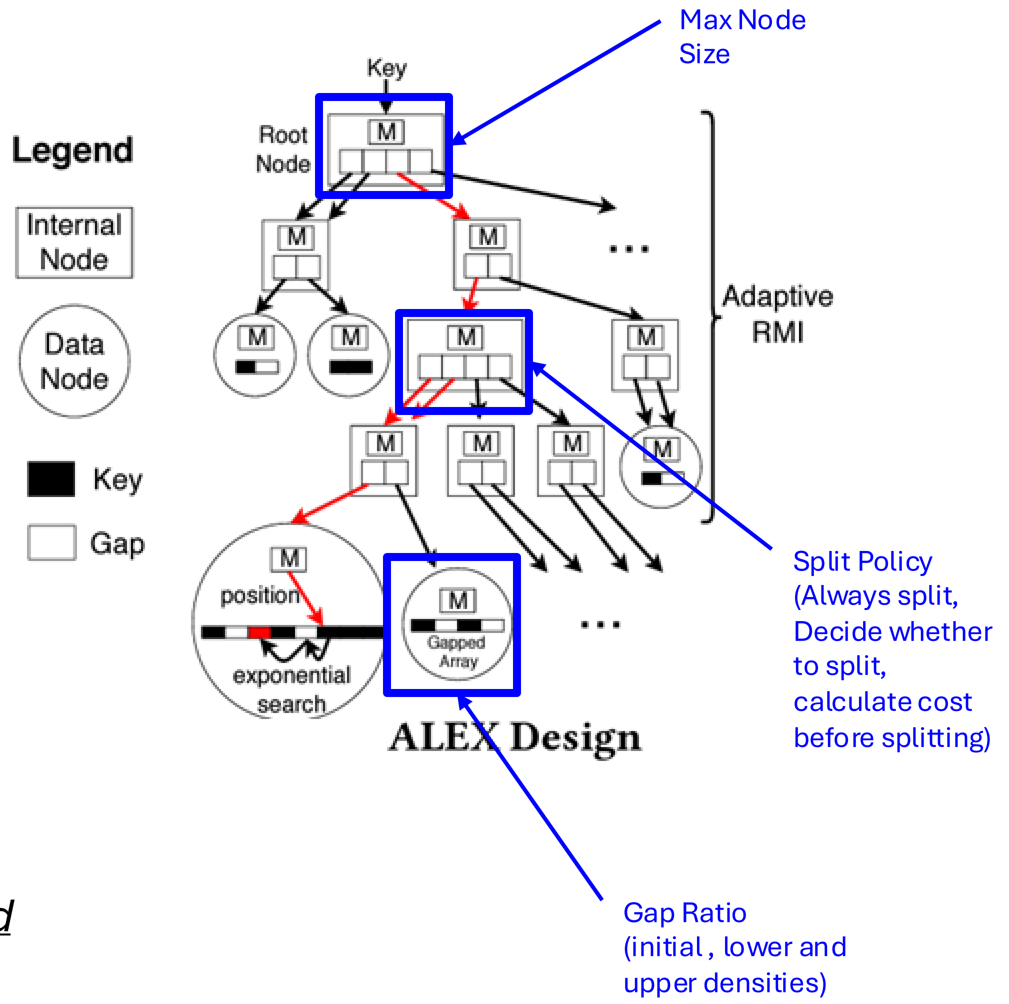
Other hidden features on Learned Index

- Highly parameter-dependency
- Different workloads require different optimal settings
- Learned Indexes are vulnerable when parameters are changed

Example:

ALEX 14-D tunable parameters vs B+Tree's very few parameters like fanout

Key Question: *How can we* ***automatically*** *find the best configured parameter settings under different workload cases (queries and data distributions)?*



Motivation: Tuning Learned Index is highly beneficial!

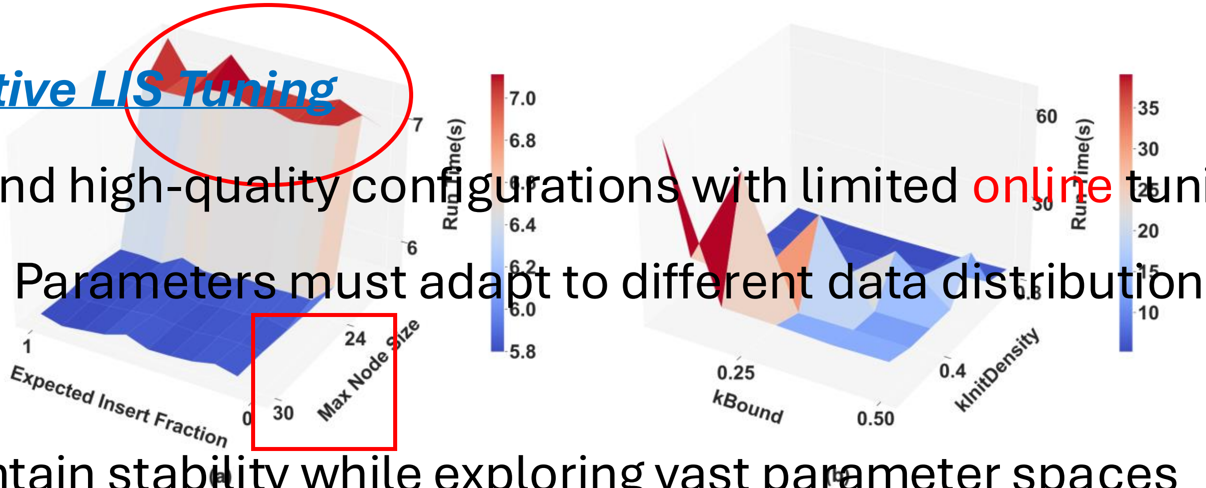
Parameter Tuning Shows Massive SLO Performance Gains

- Optimal settings can achieve 10x+ speedup over poor configurations
- Different parameter interactions create complex optimization landscape
- Current practice of parameter setup relies on manual tuning or simple heuristics by experts

Bottom charts: *Smart parameter tuning unlocks significant performance potential*

Key Challenges for Effective LIS Tuning

- **Online Constraints:** Find high-quality configurations with limited online tuning budget
- **Workload Adaptation:** Parameters must adapt to different data distributions and query patterns
- **Safe Exploration:** Maintain stability while exploring vast parameter spaces



ALEX-OSM-Balanced

Tuning Learned Index – Opportunities

Stateful

Index state metrics and structure info directly reflect workload impacts

Adaptive

Tuner may anticipate and adapt to evolving workload patterns

Long-term Planning

Sequential decision-making approach with adaptive exploration suits dynamic optimization

REINFORCEMENT LEARNING
Core Method

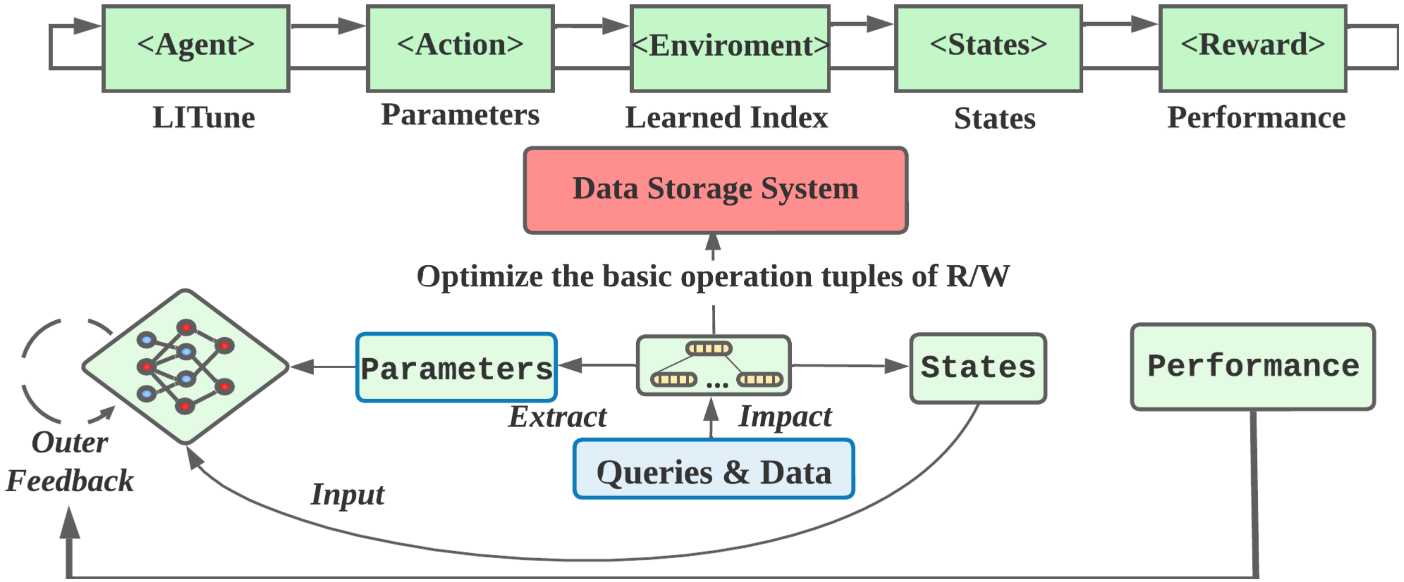
Intuitively…

Adaptive exploration within complex parameter space

Why (Deep) RL?

Generalizable representations under different workloads

RL Formulation for LIS Parameter Tuning → LITune



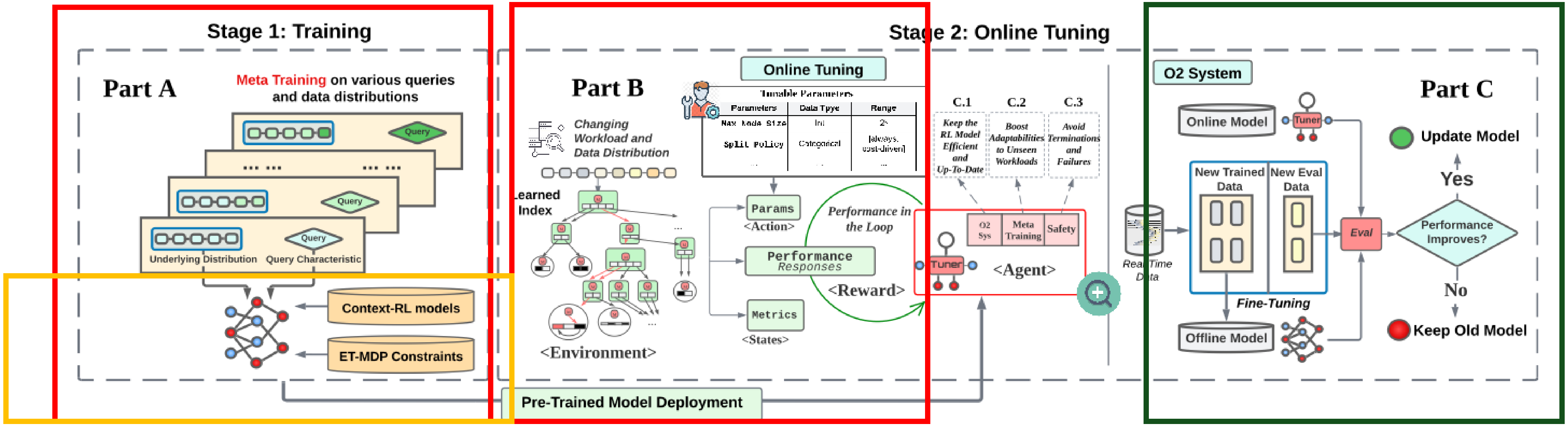
However... vanilla RL Is Not Enough for LIS Tuning, because:

- **Workload Diversity:** Rapid adaptation needed across unseen RL tasks
- **Safety Constraints:** Aggressive exploration can crash systems during tuning
- **Experience Transfer:** Previous knowledge should accelerate new workload optimization
- ***Solution:*** *Meta-RL enables fast adaptation + Safe-RL ensures stable exploration*

In one sentence, LITune is a fast, efficient and reliable online Learned Index tuner

System Overview: Features of LITune

- Feature1-Adaptive and Efficient Tuner: Meta-RL + effective representations.
- Feature 2-O2 Module: Offline fine-tuning + online tuning keep the tuner renewed.
- Feature 3-Safe Tuner: Context-aware RL (i.e. Safe RL) ensures the stable and safe tuning



Feature 1 of LITune: Adaptive and Efficient Tuning

- **Adaptive Training Module:** Meta-RL + transfer learning across diverse workloads
- **Workload Shift Response:** RL agent captures shared state transitions + React to workloads shifts

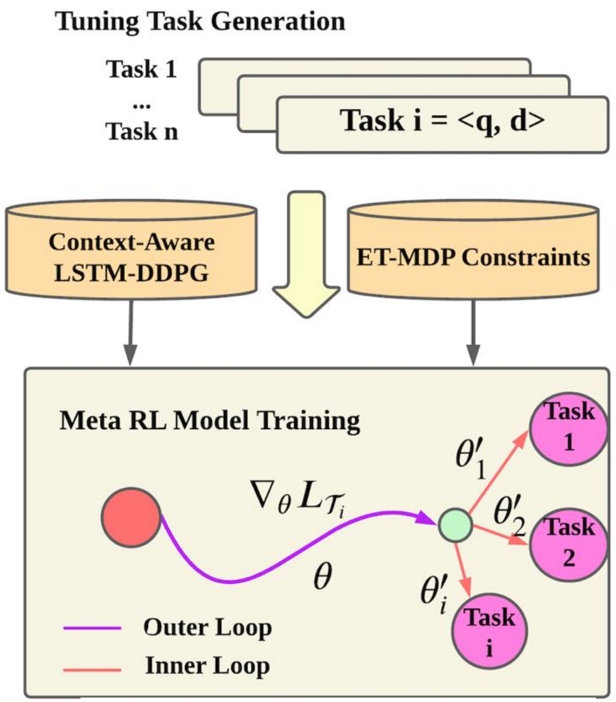


Figure: Meta-RL

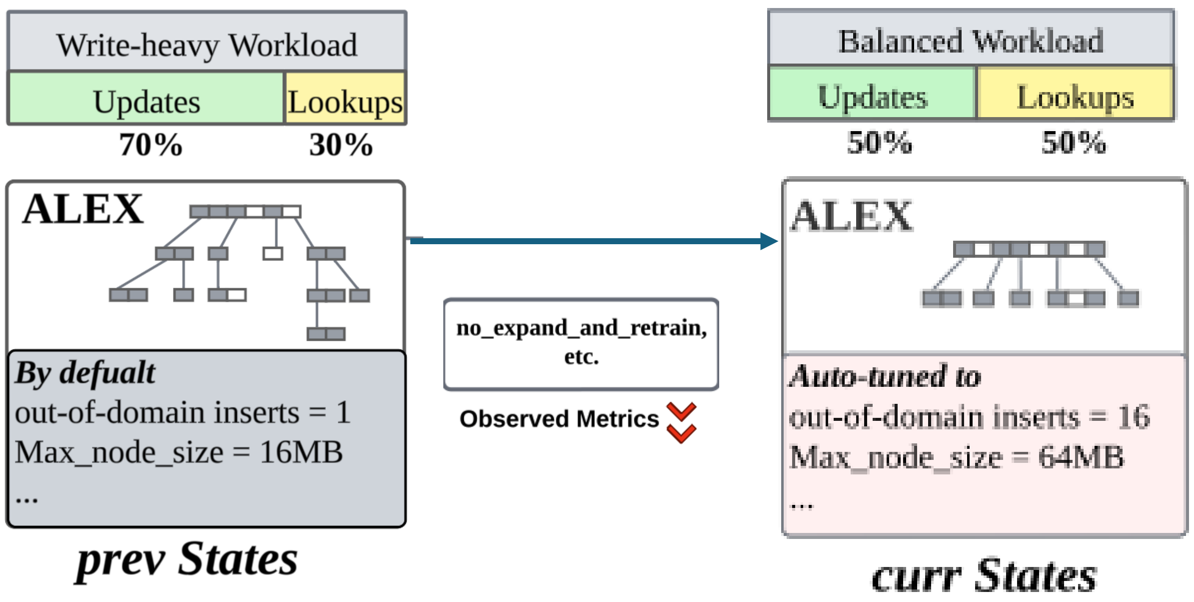


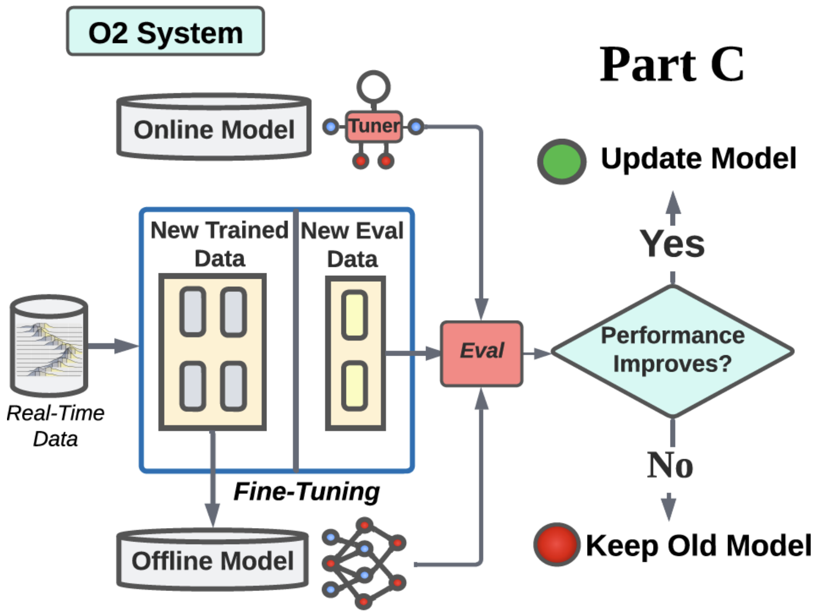
Figure: Adaptive online tuning

Feature 2 of LITune: O2 Module

Dual Model Architecture: Separate offline and online models for optimal performance

Smart Model Updates: Updates triggered only when new data improves effectiveness

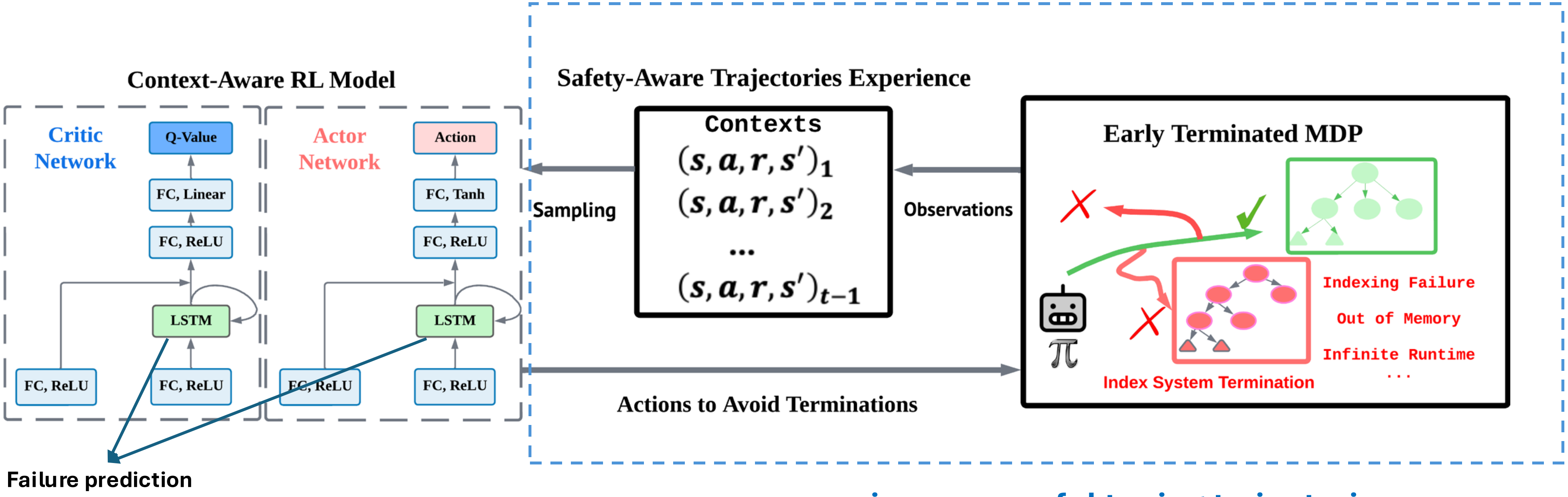
Out-of-Scope Handling: Renews models for data patterns beyond training scope



Feature 3 of LITune: Safe tuner via context-aware RL

We convert common MDP (Markov Decision Process) to Early-Terminated MDP, and introduce a **Context-aware RL model:**

- **Safe Path Learning:** LSTM module memorizes successful trajectories and predicts system failures
- **Early Termination Rewards:** Maximizes performance before hitting termination conditions



memorize successful tuning trajectories

Experimental Setup

Test Environments:

- NVIDIA Quadro RTX8000 GPU with Intel Xeon Gold 5218 CPU, 8 vCPUs, 64GB RAM
- Repeated 5 times with different seeds for statistical significance

Workloads: Search on Sorted Data (SOSD) Benchmark with different query patterns

- Static Workloads: OSM, Amazon books, Facebook user IDs, MIX distributions
- Dynamic Workloads: Continuous tumbling windows with rapid data evolution
- Variable Write-Read Ratios: Balanced, Read-Heavy, Write-Heavy

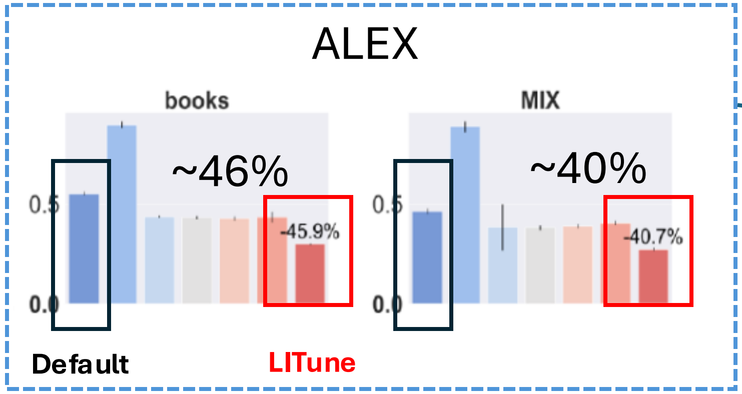
Learned Index Instances: CARMI, ALEX

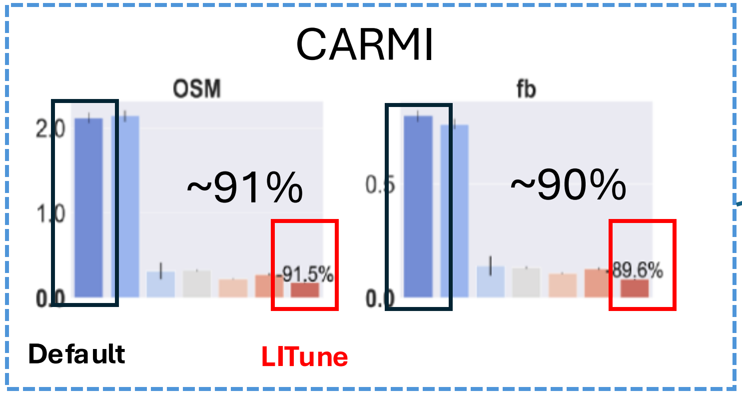
Baselines: SMBO(Bayesian Optimization), Grid Search, Random Search, Heuristic, Vanilla DDPG (RL)

Metrics: Query Runtime/Throughput/System Failure Rate

Experimental Results: E2E performance

- **Consistent Performance & Adaptability:** LITune achieves 40-90% runtime improvements across different indexes while maintaining gains across various query patterns and data distributions compared with default settings, demonstrating robustness.





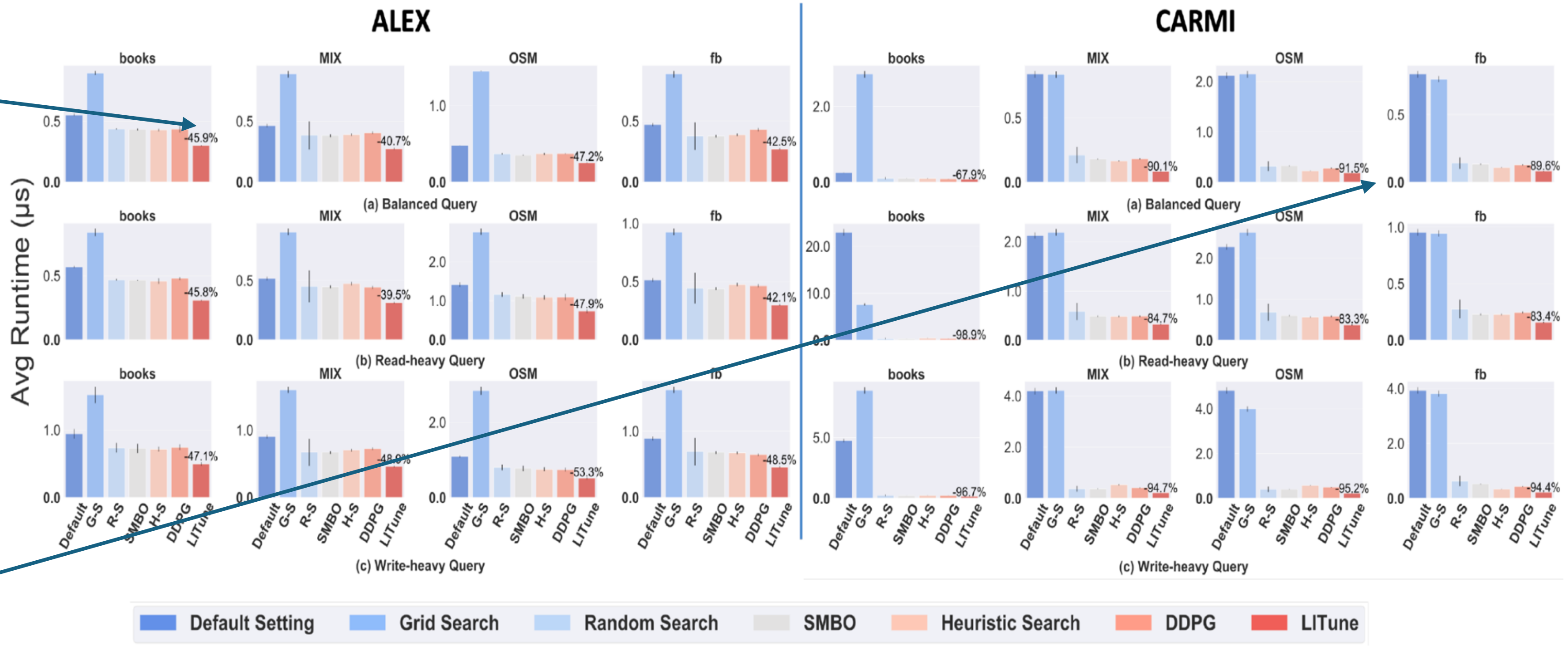


Figure: LITune End-to-end runtime performance improvement across multiple workloads

Experimental Results: Continuous and safe tuning

- **Stability Under Workload Shifts:**

Minimal performance fluctuation during workload transitions, outperforming all baselines

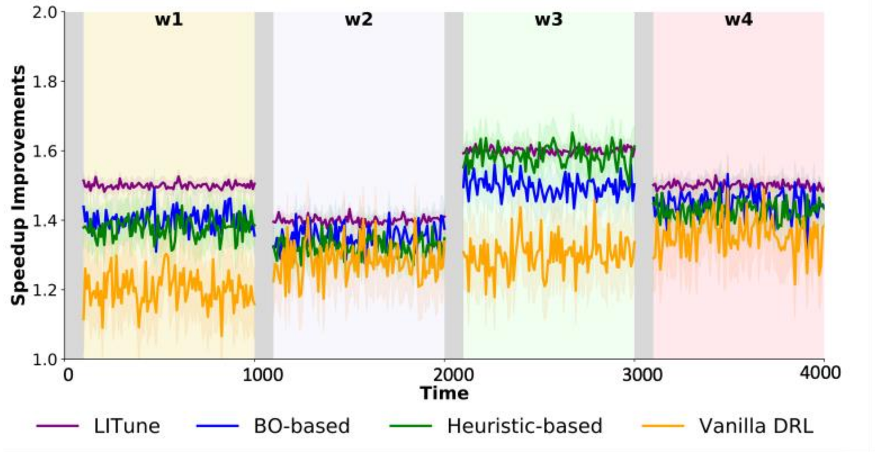


Figure: Stability under drastic workload shifts

- **Safe Exploration:**

Avoids dangerous parameter zones while other methods randomly sample risky configurations causing failures

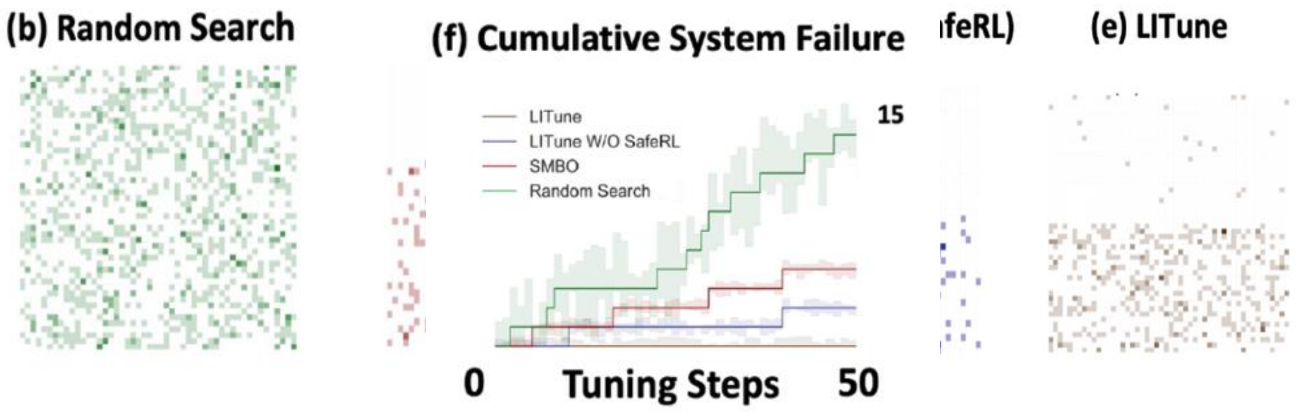


Figure: Parameter Space Exploration Safety

Conclusion

Available resources, website and demo here

Scan me!

LITune

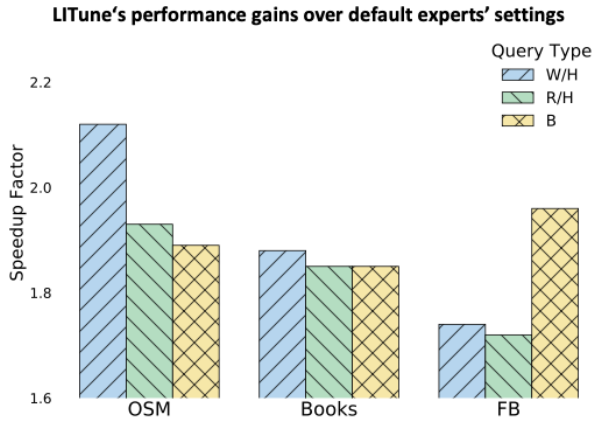
LITune: Advanced RL-Based Automated Tuner for Learned Indexes

- ✓ **Fast & Efficient Tuning:** RL-based approach enables rapid parameter optimization across diverse workloads and index structures
- ✓ **Reliable Online Behavior:** O2 module ensures continuous model updates and stable performance as workloads evolve over time
- ✓ **Proactive Risk Management:** Maintains system stability while exploring optimal configurations

LITune establishes a new online tuning paradigm for learned index performance optimization, which also provides good insights for future learned index designs.

Backup Results: Overview

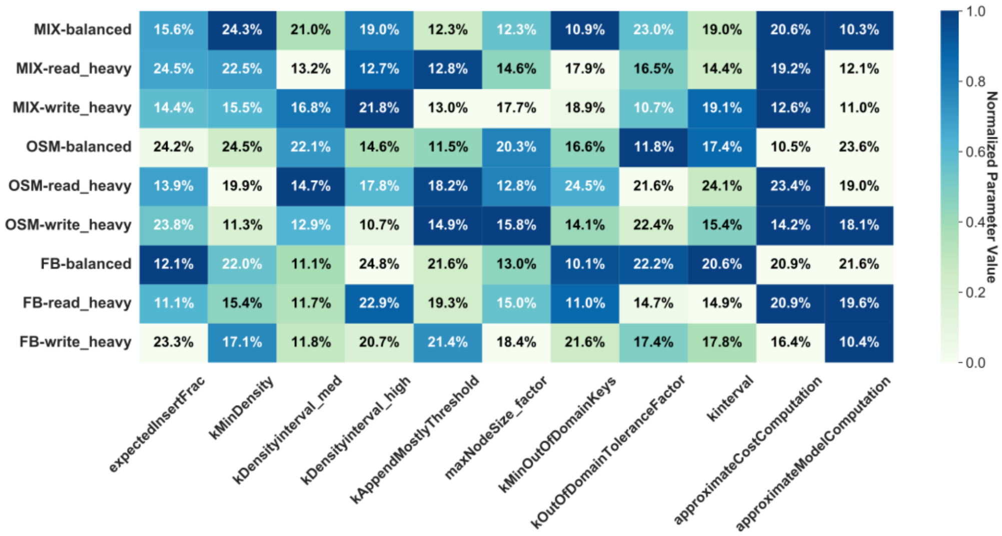
Right charts: *Smart parameter tuning unlocks significant performance potential from our preliminary experiments*



[Other Observations: Individual parameter tuning has limited impacts]()

(ALEX as an example…)

- No single parameter dominates across all workloads
- Parameter interactions are crucial for optimal performance
- Joint optimization is NP-hard



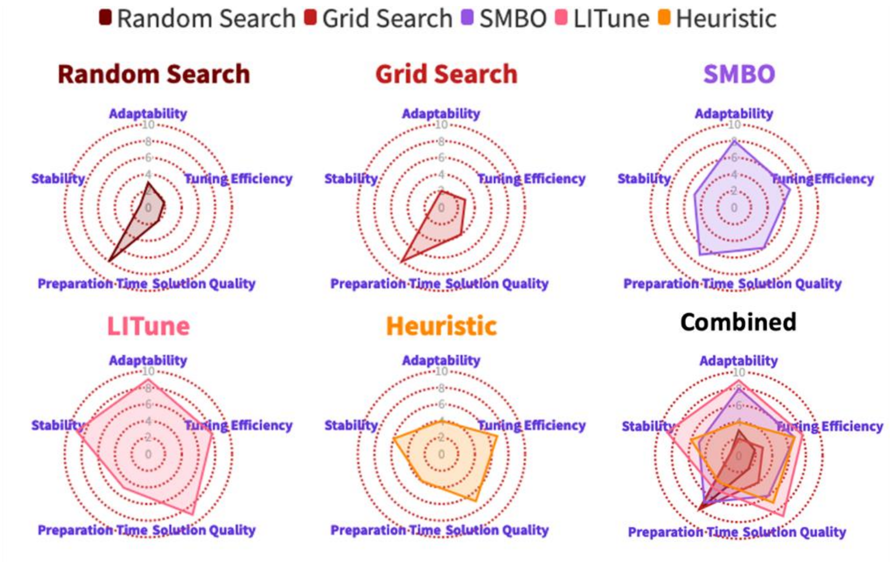
Backup Results: Overview

- **Training and Tuning Efficiency:** LITune achieves best performance with minimal tuning overhead while baselines require much longer tuning time for modest gains.

- **Multi-Dimensional Superiority:** LITune excels across all dimensions (adaptability, stability, efficiency, quality) while other methods have significant weaknesses.

Method	Training	Tuning Time				Best Perf.
		-5%	-10%	-20%	-45%	(default 403s)
Grid Search	-	32m	1.0h	-	-	→ 360s
Heuristic Search	-	15s	35s	5m	-	→ 312s
SMBO	-	12s	25s	3m	-	→ 314s
DDPG ([23, 27])	12h	28s	35s	45s	-	→ 326s
LITune-0.1%	5h	6s	12s	18s	25s	→ 288s
LITune-1%(ours)	6h	8s	15s	22s	28s	→ 212s
LITune-10%	7h	12s	20s	26s	32s	→ 211s
LITune-Full	12h	18s	25s	32s	38s	→ 208s

Table: Training and tuning overhead (LITune with different sampling-based cost models)



Radar Chat: High-level Comparison among tuning methods

Backup Results: Continuous Tuning

- **Dynamic Adaptation:** Equipped with O2 module, LITune maintains consistent performance improvements across evolving data streams, while methods without O2 degrade over time.
- **Stability Under Workload Shifts:** LITune demonstrates superior stability with minimal performance fluctuation during workload transitions (w1-w4), outperforming all baseline methods including vanilla RL approaches.

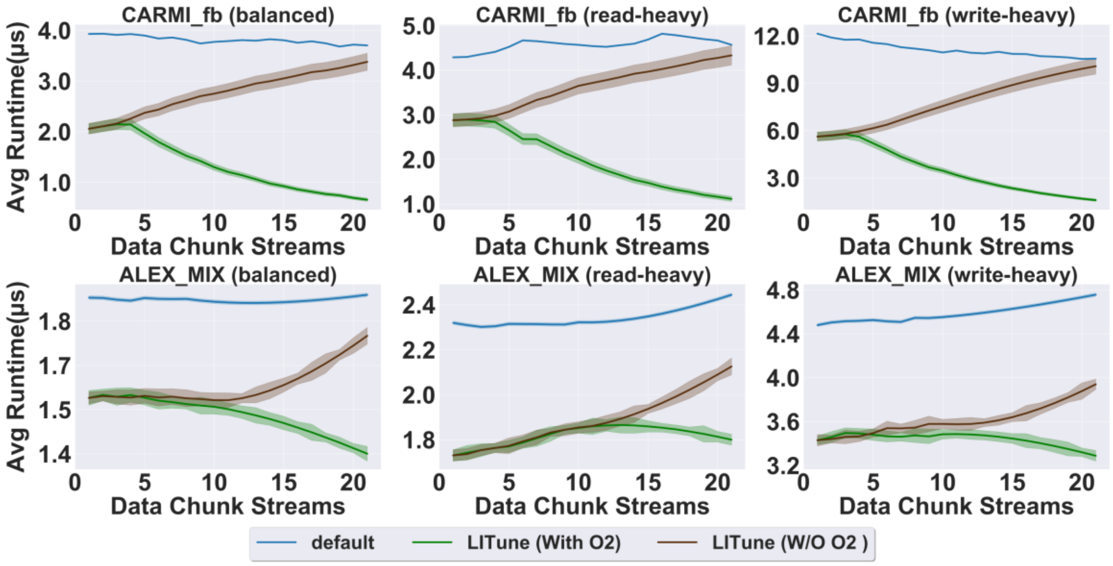


Figure: Effects of O2 on continuous Tuning

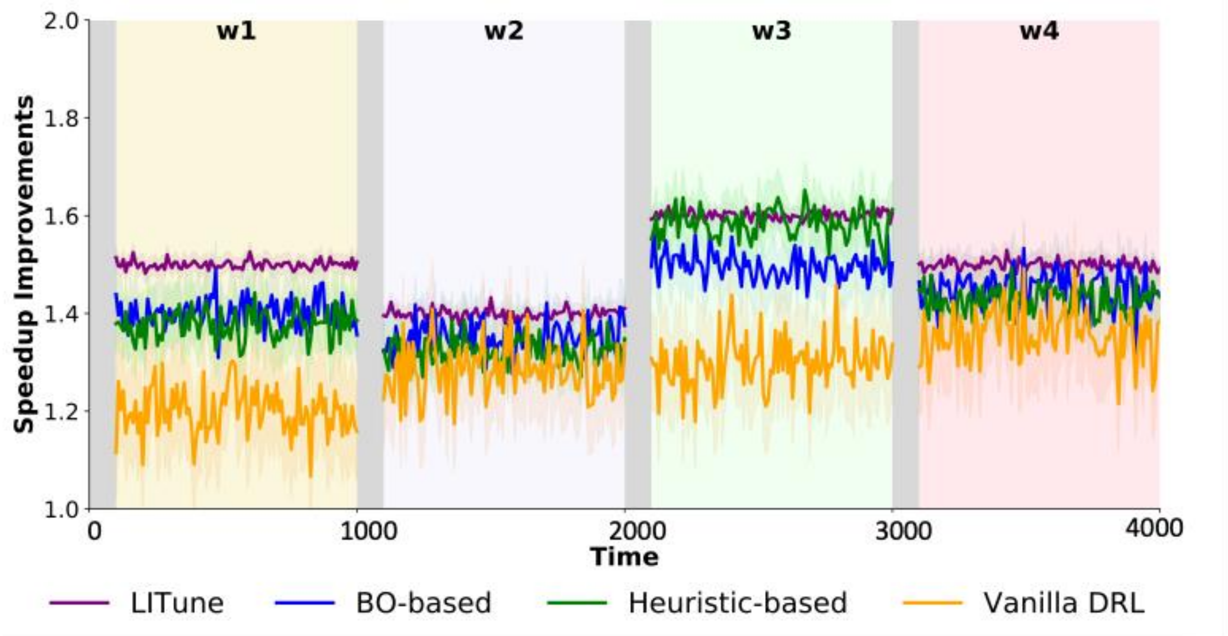


Figure: Stability under drastic workload shifts

Backup Results: Safe Tuner

- **System Safety:** LITune achieves close-to-zero indexing system crashes throughout tuning while maintaining stable performance improvements, compared to 10+ crashes from baseline methods.
- **Safe Exploration:** Our hindsight validation proves that LITune effectively avoids dangerous parameter zones and explores safe regions of the parameter space, while other methods randomly sample risky configurations leading to system failures.

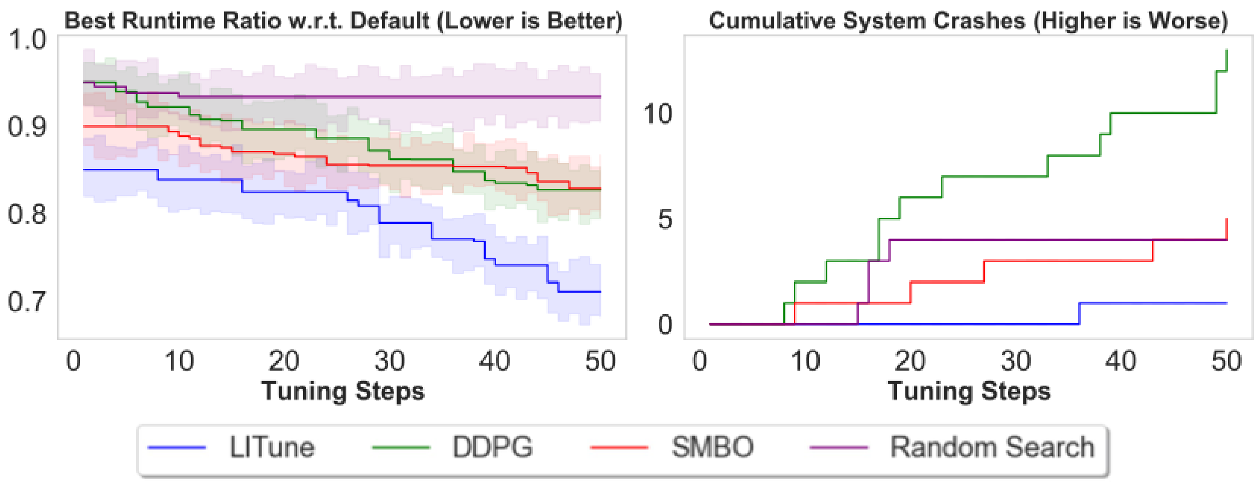


Figure: Tuning Performance and Stability

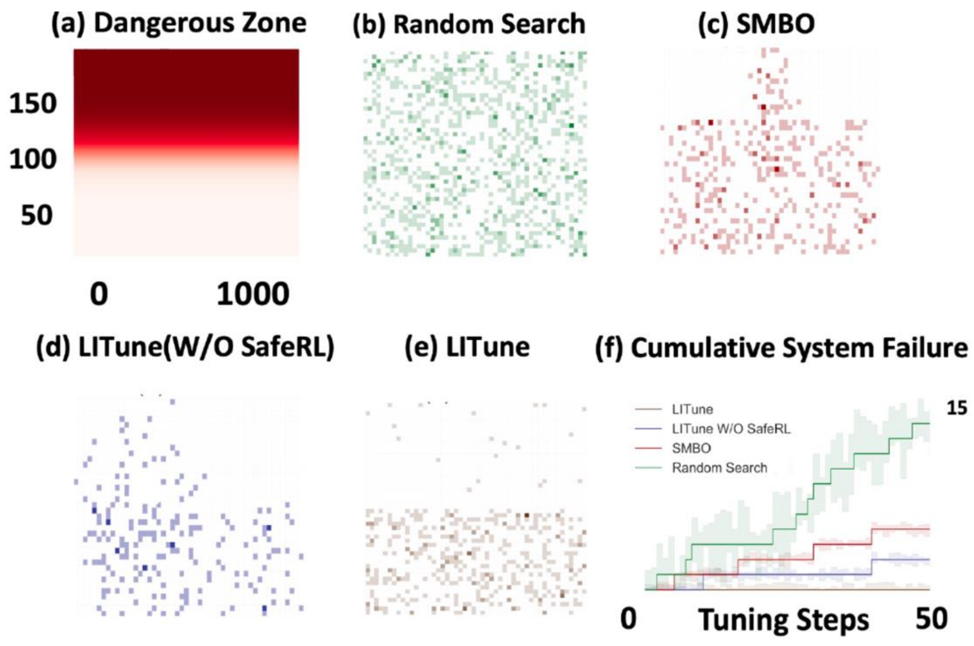


Figure: Parameter Space Exploration Safety