

Rithin Nagaraj

rithin.nagaraj@gmail.com | github.com/rithinnagaraj | rithinnagaraj.github.io

Research interests: continual learning, AI safety and control, world models for safe RL, mechanistic interpretability, state space models.

EDUCATION

PES University, Bangalore

2023 – 2027

Bachelor of Technology in Computer Science and Engineering

PUBLICATIONS AND PREPRINTS

Pharmacokinetic SSMs for Haemodynamic Collapse Prediction *AIH 2026, Best Paper Nominee (top 3%)*

Rithin Nagaraj, Sudiksha Chindula, Bhaskarjyoti Das. [DOI] [arXiv]

- Engineered Propofol and Remifentanyl effect-site concentration trajectories as explicit inputs for Mamba-based Selective State Space Models, preventing a **13.9% AUPRC drop** compared with haemodynamics-only baselines.
- Quantified lead-gap contamination in prior evaluation setups, with strict temporal boundaries reducing apparent AUROC by **16.7%** compared with the less constrained evaluation.

CAP-Adam: Inspectable Protection for Continual Learning

Under review at TMLR

Rithin Nagaraj, Paras Chopra.

- Proposed a continual learning optimizer that modulates parameter updates by their importance to previously learned tasks, outperforming AdamW and established baselines including Elastic Weight Consolidation (EWC).
- Achieved approximately **90% retention across 40 unique tasks** while maintaining **91% mean current-task accuracy over 200 sequential tasks** on Permuted MNIST with an MLP-300 model.
- Showed that optimizer hyperparameters systematically control the stability-plasticity trade-off, enabling controlled studies of knowledge retention and adaptation.

Comparing Latent Concept Formation in SSMs and Transformers via SAEs

arXiv preprint, 2026

Rithin Nagaraj, Perna Subhashchandra Kunder, Rupa Laalasa Oruganti, Aswhini Joshi. [arXiv]

- Mapped feature-level correspondences between Mamba-130m and Pythia-70m using sparse autoencoder dictionaries over **10M text tokens**, finding no evidence of architecture-exclusive Dark Matter features.
- Identified monosemantic landmark features in Mamba’s recurrent hidden state and hypothesized that they act as periodic state resets that compensate for the absence of global positional attention.

EXPERIENCE

Independent Research Collaborator

Aug 2026 – Present

Continual reinforcement learning and the cost of safety correction in LLMs.

With Maksim Anisimov, PhD candidate, Imperial College London.

Research Intern | Lossfunk

Jun 2026 – Present

Continual learning optimizers (CAP-Adam, under review at TMLR) and AI control for LLM agents.

Research Intern | RAPID Labs, PES University

Jun 2024 – Aug 2024

Vision classifiers to predict arability of soil for exotic crops using satellite imagery

ONGOING RESEARCH

Continual RLVR and Safety-Support Collapse [Code]

Imperial College London

- Asking whether training a 3B Qwen model on a sequence of harmless tasks with **RL from verifiable rewards (RLVR)** can quietly make useful, safe answers on earlier tasks unreachable, even while task accuracy and aggregate safety metrics look stable.
- Measuring how many rollouts **corrective RLVR** needs before earlier tasks give a useful learning signal again, to see whether continual training makes safety fixes more expensive.

Adversarially Trained Container Design for LLM Agent Control [Code] Lossfunk

- Training a **designer model** against LLM agents to write Docker configurations that stop a possibly misaligned agent from escaping or causing harm, without blocking the harmless work it is there to do.
- In a first sanity check with **Qwen3-30B-A3B**, **DeepSeek-V4-Pro** and **GLM-5.3** as agents (4 routes to harm, ~5,000 possible configurations), a simple softmax policy found a setup that was both safe and useful within **20 episodes**.
- Next: adding more independent routes to harm and opening up a much larger configuration space.

Confounding in Learned World Models for Anesthesia Dosing [Code] PES University

- Clinicians adjust doses in response to what they see, so a model trained on recorded surgeries can pick up those reactions instead of how the drugs actually work. Testing whether world models fall into this trap, since RL dosing policies would be trained inside them.
- Trained **25 models** (GRU, Transformer, RSSM, a 5-RSSM ensemble, and a hybrid where doses act only through pharmacokinetic–pharmacodynamic equations; 5 seeds each). The black boxes predict well: 6.9–7.2 five-minute BIS error, **27–29% better** than pharmacology alone.
- Early counterfactual tests suggest they learned only part of the story: drug effects point the right way but are just **12–21% as strong** as pharmacology predicts and arrive **3–7 min late**. The hybrid recovers ~**72%** of the expected effect with correct timing, at a 12% cost in BIS accuracy.

PROJECTS

Stable-Drift Re-implementation [Code] Apr 2026

- Re-implemented the ICCV 2025 latent drift-guided replay method on CIFAR-10, selecting exemplars by representational change under domain adaptation and validating that drift-based selection reduces forgetting compared with random replay.

Probing Compositional Limits in Object-Centric Learning [Code] Jan 2026

- Built a frozen Slot Attention plus relational graph pipeline for unsupervised object-part binding, using connected components and pairwise relations to bridge over-segmented object parts.
- Achieved **0.775 F1** on texture-flattened PartImageNet++ and used zero-shot transfer from vehicles to quadrupeds to probe how strongly learned relations depend on domain-specific latent geometry.

World Models Re-implementation [Code] Nov 2025

- Reproduced the complete VAE, MDN-RNN, and CMA-ES controller pipeline from scratch on CarRacing-v0, including random rollout collection, latent representation learning, sequence modeling, and controller optimization.

TECHNICAL WRITING

Reinforcement Learning | Grokking and Neural Fourier Circuits | Slot Attention in JAX

TECHNICAL SKILLS

Languages: Python, C, C++, Bash

ML frameworks: PyTorch, JAX, TransformerLens, NumPy, SciPy, Scikit-learn

Methods: State space models (Mamba), world models (RSSM), sparse autoencoders, reinforcement learning (RLVR), continual learning, graph neural networks

Tools: Git, Docker, Weights & Biases, Jupyter