

James Steiner

jas330@cantab.ac.uk • [linkedin.com/in/james-a-steiner](https://www.linkedin.com/in/james-a-steiner) • jamessteiner.github.io • github.com/JamesSteiner

EDUCATION

University of Cambridge

B.A. (Hons) in Computer Science, Upper Second Class Honours

Cambridge, UK

2021 - 2024

Sevenoaks School

International Baccalaureate: 45/45 points (top 1% worldwide)

Kent, UK

2014 - 2021

RESEARCH PROJECTS

VLA Action Heads: Flow Matching vs Regression vs Diffusion | Python, Pytorch

May 2026 - June 2026

Write-up: jamessteiner.github.io/projects/vla-action-heads · Code: github.com/JamesSteiner/vla

- Ran a controlled comparison of three action-generation objectives for a Vision-Language-Action policy (SmolVLA: frozen SmolVLM2-500M vision-language backbone + ~100M-param action expert), holding backbone, data (LIBERO), and training recipe fixed so only the loss/sampling differs, avoiding the architecture confound that muddies most such comparisons.
- Implemented and evaluated L1 regression, flow matching, and DDPM diffusion heads end-to-end (design, training, and closed-loop LIBERO evaluation): flow matching 73.8% and L1 regression 70.0% task success, with regression at ~2× lower inference latency (286 vs 615 ms/chunk) and smoother actions; DDPM trailed at 38.8%.
- Found the simplest objective (deterministic regression) matches flow matching on this unimodal demo data at half the cost, and scoped the claim honestly (one head, one suite, one seed), noting where generative heads should win (multimodal action distributions).

Self-Supervised World Model on Driving Video (V-JEPA-style) | Python, Pytorch, DDP

May 2026 - June 2026

Write-up: jamessteiner.github.io/projects/jepa-world-model · Code: github.com/JamesSteiner/jepa-world-model

- Implemented a V-JEPA-style joint-embedding predictive model: a context encoder (ViT-B/16, sees ~12% of patches via tube masking) predicts an EMA target encoder's latent embeddings of masked patches; smooth-L1 loss in latent space, trained with 2-GPU PyTorch DDP on nuScenes video.
- Built a controlled latent-vs-pixel comparison where the encoder is identical and only the prediction target differs (latent embeddings vs MAE-style raw-pixel reconstruction), isolating the effect of predicting semantics over appearance.
- Probed representation quality by regressing ego-vehicle speed from frozen features: the JEPA encoder reached $R^2 \approx 0.75$ (k-NN) vs a negative R^2 for the pixel baseline; diagnosed and reported a probe-overfitting failure and leak-tested the collapse-avoidance wiring (EMA + stop-gradient).

TAID: On-Device Japanese AI Code Reviewer | TypeScript, React, Python, FastAPI, WebGPU

Feb 2026 - Mar 2026

- Built six specialist reviewer-persona LLM agents (Security, Performance, Architecture, Quality, QA, Mentoring) running in parallel on a GPT-4o cloud path and sequentially on-device (TinySwallow 1.5B via WebLLM/WebGPU), behind a shared 'ReviewStreamStrategy' interface so providers swap without UI/business-logic changes.
- Designed a two-pass extraction pipeline separating free-form generation (temp 0.15, streaming) from structured JSON extraction (temp 0), solving a reliability problem a 1.5B model cannot handle in one call.
- Built the evaluation framework: scored prompt quality across 7 code snippets × 4 pipeline stages × 2 providers × 3 runs; cloud 82% PASS, on-device 77%, with failures localised to specific stages.

Periodic Attention Bias in Transformers | Python, Pytorch

Sep 2025 - Oct 2025

Write-up: jamessteiner.github.io/projects/periodic-attention-bias · Code: github.com/JamesSteiner/periodic-attention-bias

- Replaced learned absolute position embeddings with a compact periodic attention bias: a per-head sum of K=12 sinusoidal components with learnable amplitudes, frequencies, and phases over relative token distance (a Fourier-basis relative-position bias).
- Showed 3,456 bias parameters can fully substitute for 131,072 position-embedding parameters (38× fewer) while matching the baseline on enwik8 (1.312 vs 1.323 bpc); analysed the trained checkpoint, finding heads combine ALiBi-like recency decay with multi-scale periodic structure.

Neuroevolution of Augmenting Topologies (NEAT) Implementation in JAX | Python, JAX

Nov 2025 - Dec 2025

- Implemented NEAT from scratch in JAX (genomes as adjacency matrices, 'jax.lax.scan' over topological sort for JIT-compatible forward passes); applied to SlimeVolley self-play, and extended to Backprop NEAT using JAX autodiff for weight training within evolutionary topology search.
- Identified and characterised an emergent reward-hacking strategy (the population found a match-timeout loophole in the fitness proxy), a hands-on case study in Goodhart's Law and the importance of objective design.

Stable Diffusion Fine-Tuning for Kanji Generation | Python, PyTorch, Diffusers

Aug 2025 - Sep 2025

- Constructed ~69,000 text-image pairs from 6,410 kanji (KANJI2 meanings + KanjiVG glyphs), using meaning-based and radical-aware captions to ground compositional structure in CLIP's embedding space.
- Fine-tuned the SD 1.5 UNet (LR 1e-5, AdamW, 80k iterations, batch 4 × grad-accum 4); compared full fine-tuning against a rank-4 attention-only LoRA baseline: the LoRA produced noise, while full fine-tuning yielded structurally plausible glyphs on novel prompts.

Vector Runahead in CPU Simulator | C++, Python

Sep 2023 - May 2024

University of Cambridge, supervised by Prof. Timothy Jones

- Reproduced and extended the Vector Runahead architecture paper in gem5 (a cycle-accurate CPU simulator), implementing commit-stage pseudo-retirement, dependency/taint tracking, and a three-state pipeline mode machine; a 9-month supervised research project taken from paper to working, measured implementation.
- Reduced L2 cache miss rates by up to 12% across 5 graph workloads, demonstrating low-level performance-optimisation depth relevant to compute- and inference-aware ML implementation.

PROFESSIONAL EXPERIENCE

Clarion Housing Group, Software Developer

Aug 2024 - present

- Introduced AI-assisted development tooling: integrated the Chrome DevTools MCP server so an LLM agent can inspect running pages during development, and authored a reusable "feature run-through" agent skill that reproduces new features and logs platform-specific issues.
- Built full Azure DevOps CI/CD from scratch for a new internal tool, and owned pre-merge CI on two React/Next.js production sites (linting, formatting, >90% coverage across Jest, Playwright, Cucumber), giving reproducible, tested delivery.
- Wrote a custom esbuild `onLoad` plugin to integrate the React Compiler's Babel transform into a tsup build where no public integration existed.

Nokia Bell Labs, Student Software Developer Intern

Feb 2023 - Mar 2023

- Built a real-time AI safety/fairness evaluation tool for ML notebooks: auto-scanned code and datasets, integrated Microsoft Fairlearn for bias metrics, and used a context-extraction pipeline before querying GPT-3/Codex for analysis and fix suggestions.
- Iterated prompt design to improve detection metrics; ran biweekly feedback with an AI-ethics client and a 6-person team

AWARDS

UKMT Hamilton Olympiad, Silver Medal (ranked top 50 nationally)

UKMT BMO (British Mathematical Olympiad), distinction (ranked top 200 nationally)

UKMT Senior Mathematics Challenge, Gold Award (scored 120/125)

UK Chemistry Olympiad, Gold Medal (ranked top 500 nationally)

SKILLS & INTERESTS

Technical Skills: Python, C/C++, TypeScript, JavaScript, SQL, Git, Linux, Docker, AWS, FastAPI, React.js, Next.js, PyTorch, HuggingFace Transformers, PEFT/LoRA, Diffusers, ViT / DINOv2, VLA fine-tuning, diffusion & flow matching, self-supervised learning (JEPA), transformer architecture, LLM agents & evaluation frameworks; distributed training (PyTorch DDP); experiment tracking (Weights & Biases); JAX, TensorFlow.

Languages: English (native), Japanese (intermediate reading ~N3, basic conversational; studying towards N2), Mandarin (intermediate), French (intermediate).