AKBAR JURAEV

Human-Computer Interaction & AI Researcher · Robot Learning & Embodied Systems

Birmingham, UK | hello@akbarjuraev.com | akbarjuraev.com | linkedin.com/in/akbarjuraev | github.com/abyyworld

Researcher and engineer reading Artificial Intelligence & Computer Science at the University of Birmingham. My research sits where human-computer interaction meets AI systems: immersive and multi-device interaction, human-agent collaboration, and how people stay in control of AI behaviour, with four studies in preparation for CHI and IEEE VR. In parallel I build the infrastructure around robot learning: reproducible benchmarks, demonstration pipelines, policy evaluation, and fleet deployment. Educated at Presidential School in Bukhara, part of Uzbekistan's most selective school network.

RESEARCH

Error Feedback in Immersive Voice Programming (DreamCodeVR)

University of Birmingham · Wizard-of-Oz user study, manuscript in preparation for CHI 2027

When an AI writes your code and it breaks, can you still tell whose mistake it was? In VR, speech is the only tool you have to find out. Research study investigating how users understand, attribute, and recover from AI-generated code errors in virtual reality. The underlying system turns spoken commands into Unity C# through an LLM and compiles it live in-scene; when generation fails, users must work out what went wrong and how to fix it through speech alone. The study compares three controlled feedback conditions (no feedback, an explanatory text panel, and an embodied conversational agent) to measure their effect on error attribution, recovery strategies, and trust. I contributed to the idea, implemented most of the technical work, and am now contributing to the study design. The infrastructure is a Wizard-of-Oz architecture in which a hidden researcher injects scripted outcomes in real time across four tasks and five outcome types (including gradually-revealed and visually-subtle errors), condition management, a researcher control interface, live speech transcript display in VR, and per-participant event and questionnaire logging. Stack: Unity/C#, Node.js, WebSocket/UDP networking, speech-to-text integration. **GitHub**

Omni-Connect / DeviceSphere

University of Birmingham · multi-device HCI research system, targeting CHI 2027

Close your fist over a function on the laptop, open it toward the tablet, and it is there: three devices, one workspace, no installation. Cross-device interaction research in which a laptop, phone, and tablet operate as a single connected workspace: the laptop runs a native Unity app doubling as the coordination server, while the phone and tablet join through zero-install WebGL clients served over the local network. Implemented a novel cross-device copy-paste interaction driven by hand gestures captured with Meta Project Aria 2 research glasses: closing a fist over a function tile grabs it (the target device previews the incoming paste), and opening the fist toward the device pastes it. Built the real-time WebSocket-to-UDP bridge, gesture-streaming pipeline with automatic reconnection, command-line Unity build tooling for four client builds, and the full infrastructure for two controlled user studies (a cross-device target-selection task and a multi-device creation task) with per-participant CSV logging. **GitHub**

AgenticXR: Safe Agentic Authoring Inside Immersive Environments

University of Birmingham · builds on DreamCodeVR, originally developed at UCL · targeting IEEE VR

Letting an AI edit the world you are standing in is only safe if it can be tested somewhere else first: generated code runs against a hidden clone of its target before it is allowed to touch the live scene. Research system in which a person wearing a VR headset creates interactive behaviour by speaking, while AI agents on a server generate, critique, and validate that behaviour before it is allowed to change the world the person is standing in. The contribution is a runtime architecture for keeping a live human and a slower, asynchronous agent system coordinated safely: an XR runtime owning everything embodied (speech, gaze, selection, preview, confirmation, undo) against an agent runtime owning deliberation, joined by a typed message contract with correlation IDs, scene epochs, and object revisions. Three mechanisms hold it together: a persistent Shared XR Memory (occupancy, semantic scene graph, script state, temporal history, consent), a Verification Space in which generated code is compiled and run against a hidden clone of the target object before anything touches the live scene, and five graded levels of agent initiative and consent matched to the risk of each action. Responsible for evaluation and deployment: designed the within-subjects study (five tasks, four hypotheses, NASA-TLX, SUS, IPQ presence, human-automation trust), built the study harness and researcher control panel merged upstream into the main research repository, solved Quest deployment, push-to-talk and zero-configuration LAN discovery, brought the agent backend up on Apple Silicon, and ran the first live speech-to-text validation of the system. Stack: Unity 6, Meta Quest, Ubiq, C# with Roslyn runtime compilation, Node.js, Claude Agent SDK, MCP, faster-whisper.

LLM-Designed Affective VR Environments

University of Birmingham · MSc research collaboration · VR & affective computing

Can an LLM design a room that reliably makes you feel calm, and does the shape of the room change the answer? Study of whether an LLM, constrained to a fixed pool of interior-design parameters, can compose virtual rooms that reliably induce a target emotion in the person standing in them, and whether room shape moderates that effect. Four emotions are drawn from the diagonal quadrants of Russell's circumplex model, within-subjects, crossed with two room geometries. Contributed to ideation and research

meetings, and built the technical system. The Python pipeline is a single-source-of-truth architecture in which the parameter pools are defined once as data and the prompt text, JSON output schema, validator, control arm, and Unity C# constants are all derived from it, with the constants file code-generated and a test that fails if engine and pipeline drift apart; enum-bounded structured LLM output; a three-layer validation gate ending in C# at scene load, since a config hand-edited on the headset never passes through Python; and a reject-and-re-ask loop that feeds violations back verbatim and retains rejected candidates, so constraint-violation and duplicate-combination rates survive as experimental results rather than being discarded. 73 automated tests requiring no API key or network, plus an adversarial fixture in which every entry must fail for a different documented reason. The Unity side procedurally generates both room shells, including a semicircular vault mesh Unity has no primitive for, so that matched entrance width, depth, ceiling height, standing position, and both sightlines are machine-verified rather than eyeballed, with a runtime material system driving hue, saturation, and roughness under a luminance-preserving albedo correction and a lighting rig exposing illuminance as the single lighting variable. Stack: Python 3, Claude API (structured outputs), JSON Schema, C#, Unity 6 / URP, Meta Quest.

LLM Bias Mini-Evaluation

Independent research

Reproducible evaluation of directional gender bias across BERT, RoBERTa, and GPT-2, combining masked-language-modeling probes with stochastic open-generation runs (10 stereotype-style prompts, multiple runs per prompt). Found a consistent directional association: across all masked prompts, the top predicted token skewed to one gender in both BERT and RoBERTa, while generation results varied across runs, suggesting instability in how bias surfaces. **GitHub**

ROBOT LEARNING & ML INFRASTRUCTURE

Correlated Perception Error in Semantic 3D Mapping

Independent research · reproducible benchmark · Docker, hash-locked dependencies, CI-enforced claims

A 70%-accurate labeller with uncorrelated errors builds a better map (0.560 mIoU) than an 85%-accurate labeller whose errors repeat by viewing angle (0.502), so the standard accuracy benchmark for a perception model can rank two models backwards. Semantic mapping systems fuse per-frame segmentation labels into a voxel map using Bayesian updates, which assume each observation of a voxel is an independent opinion. Real cameras violate that assumption: a surface voxel is visible from only one side, so it is observed from a narrow band of viewpoints, and segmentation models fail consistently from particular viewpoints, so the same mistake is counted as repeated independent evidence. I built two labellers over an identical room, camera trajectory, and ray set to isolate this. Holding accuracy fixed at 85% and varying only error correlation isolates the effect: a 0.115 mIoU gap and 4× worse calibration error. The damage concentrates on small objects (~20% loss on tables and chairs), seen from the narrowest range of angles and exactly what a robot needs the map for. Shipped with seven CI checks on every push: four validate the benchmark itself (a perfect labeller must score 100%, a random one at chance, fusion must beat single-shot, the camera must actually observe the room) and three recompute the headline numbers from scratch, so the README cannot drift from the result. Runs anywhere in ~3 seconds, CPU only. **GitHub**

Teleoperation Demonstration Pipeline

Independent project · versioned demonstration data, quality-graded datasets, reproducible policies

Nine signal-level checks identify a poor demonstration from the numbers alone, with no video review. Seven-stage pipeline taking raw teleoperation sessions to a trained, traceable policy: ingest, validate, score, dataset, train, evaluate, report. The checks detect operator hesitation (near-zero joint motion over a window), commands saturated at their limits, tracking divergence between commanded and achieved state, gripper chatter, and dropped frames, among others. Each session receives a score and a gold / silver / reject grade, and rejects never reach training. Validation is deliberately separated from scoring: validation asks whether a recording is well-formed, scoring asks whether it is a good demonstration. Dataset splitting is session-aware, so near-duplicate trajectories from one sitting cannot straddle the train/test boundary, and every trained model is pinned to a data revision and a commit, so any reported number can be regenerated months later. The scorer is validated against a generator simulating four operators of differing skill without telling the scorer who is who; it recovers the correct ranking, and a regression test fails if it ever stops doing so. **GitHub**

Policy Evaluation & Statistical Comparison Harness

Independent project · does the new model actually beat the old one, or did it get lucky?

A policy that looked 78% better than a do-nothing baseline was 27% worse than repeating the previous action, a rule that fits on one line; without that second baseline the 78% would have been the reported result. Companion system that decides whether a candidate policy is genuinely better than the incumbent. It runs both on identical seeded scenarios and returns three verdicts rather than two: better, worse, or underpowered, the last carrying the number of additional episodes needed to settle it. A six-point success-rate gap over fifty episodes sits comfortably inside sampling noise. A second finding: a policy with excellent offline metrics (very low action error, 98% gripper accuracy) achieved zero successes in 240 closed-loop rollouts: never conditioned on target position, it had learned to imitate the shape of human motion without any representation of its purpose. The offline metric was not wrong; it was answering whether the output looked human, not whether it accomplished anything. **GitHub**

Over-the-Air Policy Updates for Edge Robots

Independent project · safe model delivery to devices that move physical hardware

Update mechanism for deployed robots, built around the premise that a bad software update on a robot moves an arm rather than crashing an app. A new policy version is staged to a scratch location, proven loadable and within its latency budget on that specific hardware before it is allowed near the control loop, then switched in atomically with rollback on failure. Rejections are recorded with their reason and persisted, so a fleet does not sit re-downloading a known-bad release indefinitely. Paired with device telemetry reporting inference latency, failures, and flagged edge cases to a central store, turning an optimised on-device model into the down-link and up-link of a fleet loop.

Fleet Learning Loop

Independent project · multi-node continual improvement with canary rollout

Over eight rounds a three-node fleet moves from 32% to 94% success while uploading only 16% of the data it collects. End-to-end fleet MLOps loop at small scale: three nodes running in different conditions, each executing the shipped policy, scoring its own attempts locally, and uploading only the episodes worth uploading under a bandwidth budget. The server ingests, validates, and retrains; the new version reaches a canary subset first and is promoted to the rest only if their measured results improve, rolled back otherwise. The starting policy scores 87% on the condition it was trained for and 0% on another. The case that motivates the architecture is the rollback one. A regressed model passes every server-side gate, loading correctly, meeting its latency budget, and scoring 100% in the server's own simulator, and is caught only once the canary nodes run it.

SOFTWARE & INDEPENDENT PROJECTS

Cicatrixa

Independent project · autonomous deploy-and-heal platform

AI system that takes any GitHub repository, determines how to build it, wires a subdomain, and verifies the deployment end-to-end before reporting success. It then stays on call: every push triggers a redeploy, and every crash is diagnosed and repaired automatically, typically before the developer is aware of the failure. Live at **cicatrixa.com**.

Role Radar: Academic & Career Opportunity Tracker

Independent project · continuously verified job graph with local AI CV tailoring

Automated tracker that watches community boards alongside official Greenhouse, Ashby, and Lever feeds and maintains a continuously re-verified set of open postings (currently ~700, including research, PhD, and postdoc positions), classified by category, region, term, degree evidence, and work-authorisation status, with unknown sponsorship data marked review-required rather than assumed eligible. Surfaced through a filterable dashboard; each posting has a one-click editor that proposes evidence-checked, job-specific wording patches to a master CV and exports a tailored PDF. Privacy is part of the design: the editor runs on localhost, the public repository never receives the profile, fact bank, API key, or generated documents, and the system never submits an application. **Dashboard** · **GitHub**

Study Builder: User-Study Instrument & Records Platform

Independent project · research tooling

Platform for composing and hosting user-study instruments, built after repeatedly hitting the ceiling of general-purpose form tools in real study work: support for question types that generic spreadsheets and form builders do not express, per-study customisation, and a persistent record of studies, their live participant links, and their collected responses in one place.

EXPERIENCE

Student Researcher

May 2026 – Present

University of Birmingham · Internship

Conducting academic research in AI and human-computer interaction within a university lab, including the DreamCodeVR error-feedback and AgenticXR studies above: experimental design, study implementation, data collection, and analysis, applying machine learning and statistical methods to live research questions alongside academic supervisors.

Software Engineer

Jun 2026 – Present

Turin Polytechnic University in Tashkent · Internship

Developing a drone-based computer vision system for wind-turbine health analysis: training models to classify turbine conditions and detect operational modes from aerial imagery. Own the pipeline from dataset preparation, cleaning, and preprocessing through model training and evaluation to a production-ready, exportable model, contributing to the project from its earliest stages.

Ambassador of IT Exports, Uzbekistan-UK

Dec 2024 – Present

Ministry of Digital Technologies of the Republic of Uzbekistan

Official government-appointed ambassador facilitating IT export relations between Uzbekistan and the United Kingdom: promoting cross-border technology collaboration, representing Uzbekistan's growing tech ecosystem to UK partners, and building bridges between the two countries' digital industries.

Information Technology Intern
Aug 2025

IT Park Uzbekistan

Interned at one of Uzbekistan's leading technology hubs: managed IT operations and built institutional partnerships, supporting international companies' onboarding into the Uzbek market.

Ambassador, Birmingham International Academy
Oct 2025 – Jun 2026

Birmingham International Academy, University of Birmingham · Part-time

Represented BIA students as an official ambassador; delivered welcome talks, spoke at intake events, and organised community activities supporting international students' transition into university life.

English & Mathematics Tutor
Sep 2024 – Jan 2025

SAT Tashkent College Prep · Part-time

Tutored SAT students and mentored them through admissions to top universities abroad; co-designed test questions and practice materials for the English department.

EDUCATION

BSc Artificial Intelligence & Computer Science
Sep 2025 – Jun 2028

University of Birmingham

First-Class performance overall in Year 1.

Foundation Year, AI & Computer Science
Jan 2025 – Jul 2025

University of Birmingham

Grade: 80%+. Completed with certification.

High School Diploma, Computer Science
Oct 2021 – Jun 2024

Presidential School in Bukhara

Uzbekistan's most selective school network: state-funded boarding schools with entrance testing designed by Cambridge Assessment (roughly 50-64 candidates per place). Admitted at Grade 9 ranked 3rd of ~600 regional applicants for 24 places and 12th nationally among several thousand candidates; one of 168 students school-wide. International Cambridge pathway: IGCSE and A-Level curricula with Cambridge, Oxford, and Pearson materials, taught by international faculty. Student Council member.

HONOURS, AWARDS & TEST SCORES

Informatics Olympiad (Competitive Programming): 1st place at the school qualifying stage in two consecutive years (Grades 9-10); went on to compete at the Bukhara Regional Informatics Olympiad and in major national competitive-programming competitions that serve as indirect IOI qualifying rounds.

Bukhara Regional English Olympiad (regional education-ministry competition drawing the strongest students across Bukhara region): selected as the school's representative; 3rd place (Grade 11) with a certificate, monetary award, and early university admission offers, qualifying for the Republican (national) stage; 4th place (Grade 9) after ranking 1st at the school stage.

SAT: 1500 (Dec 2023): top-tier global performance in mathematics and evidence-based reading & writing.

IELTS: 8.0 (Apr 2024): near-native English proficiency across all four skills.

SKILLS & LANGUAGES

AI / ML	Machine Learning, Deep Learning, Computer Vision, NLP, Imitation & Robot Learning, Model Evaluation & Calibration
ROBOTICS	Semantic Mapping & Sensor Fusion, Closed-Loop Policy Evaluation, Demonstration Data Pipelines, Edge Inference & OTA Deployment, Fleet Orchestration
XR	Unity 6 / URP, Meta Quest, Project Aria, WebGL, Runtime C# Compilation (Roslyn)
SOFTWARE	Python, C#, Java, Node.js, PyTorch, REST API Development, Docker, CI/CD, Git
RESEARCH	Experimental Design, Wizard-of-Oz Studies, Statistical Inference & Power Analysis, Standard Instruments (NASA-TLX, SUS, IPQ), Data Analysis
LANGUAGES	Uzbek (native), English (fluent), Russian and Tajik (conversational), Arabic (elementary)