Matt McManus

New York, NY — mattmcmanus41@gmail.com —
mmcmanus1.github.io/research

Research Summary

I came to research through agents. *As a freshman*, I built a bot for the MIT Pokerbots Competition and watched it adapt against other agents; seeing learning unfold in a multi-agent setting made reinforcement learning feel concrete and set the tone for my work: frame problems as sequential decisions with feedback, and weave structure—physics, graphs, symbols, and time—into learning so behavior is legible and reliable. *Later, as President of MIT Pokerbots*, I deepened that focus on RL and multi-agent strategy. Coursework in Representation, Inference, and Reasoning in AI (6.S058) and Computational Thinking with Julia (6.S083) reinforced that perspective, pairing planning under uncertainty with software practices (automatic differentiation, matrix calculus, parallel/GPU computing) that make research reproducible at scale.

That RL-centric lens led naturally to work with **Prof. Una-May O'Reilly** (CSAIL/ALFA). I built a graph-structured cyber-defense simulator and cast defense as sequential decision-making, benchmarking RL baselines across network topologies and information regimes with ablations and failure analysis. Seeking interpretability, I prototyped neuro-symbolic decision layers to distill policies into human-readable rules; after head-to-head tests, we pivoted when simpler RL matched performance—an instructive negative result about when symbolic structure helps. In parallel, I evaluated large language models as decision aids for defenders on graph-based tasks (anomaly triage, vulnerability identification, attack-path planning), finding GPT-3 can chart high-quality paths with predictable failure modes. [Cyber Defense PDF]

The same "structure + learning" idea motivated my M.Eng. with **Prof. Alan Edelman** (CSAIL/Julia Lab). I developed a physics-informed neural ODE for strapdown inertial navigation in GPS-denied settings, embedding kinematic and sensor-error structure directly in the model and training against simulated and real IMU trajectories. In walk-forward tests we cut 3D position RMSE by $\sim$60% versus a tuned EKF. I built a Julia-based simulation/evaluation harness (sensor grades, trajectories, bias drift) and CI-backed pipelines for reproducibility. Encoding known physics improved out-of-sample accuracy and made the results easier to interpret. [Thesis]

To understand how constraints shape representations irrespective of domain, I also studied model capacity and training bias. In *Low-Complexity Interpolation for Deep Neural Networks*, we biased training toward small-norm interpolating solutions and observed lower test error and smaller weight norms, alongside simpler intermediate features—evidence that constraining complexity can make representations more stable and easier to analyze.

Complementing this, I asked which computations modern sequence models carry out internally. In *How Do Transformers "Do" Math?*, we tested whether a transformer trained for linear regression represents the intermediate slope w. Linear probes revealed robust encodings of w in hidden states, and reverse probes plus representational interventions that set $w \to w'$ shifted predictions as expected—causal evidence that the model uses a specific internal variable to perform the computation. Together, these studies point toward *steerable internals*: features we can identify and manipulate to enforce behavior, not just measure after the fact.

Industry roles have been a proving ground for these principles. At Two Sigma (part-time), I developed factor-neutral cross-sectional signals, designed feature- and learner-level decorrelation (orthogonalization; correlation-penalized loss), and validated with leakage-controlled walk-forward pipelines (rolling normalization; OOS rank IC/IR). At Bridgewater's AIA Labs, I design evaluation and calibration systems for LLM-driven analysis and cast prompt optimization as RL under explicit temporal/leakage constraints—building offline counterfactual replay with calibrated reward models, lightweight online A/Bs, and FastAPI/Kubernetes services so experiments remain traceable and reproducible. Across these settings, the core questions stay the same: formalize assumptions (time, leakage, provenance), inject structure to shape learning, and assess reliability with intervention-ready metrics.

Across RL, SciML, and interpretability, my aim is consistent: integrate structured priors with data-driven learning to produce transparent, reliable systems whose internal computations can be probed and steered.