

WiP: TIE Grammar: Stratified Generative Pipeline for Training Generalizable Deep Reinforcement Learning Cyber Agents

Bahirah Adewunmi, Dr. Sanjay Purushotham
University of Maryland Baltimore County

Abstract

Developing robust **Deep Reinforcement Learning (DRL)** agents for cyber security is fundamentally hampered by their inability to generalize due to the non-Independent and Identically Distributed (non-I.I.D.) nature of network topologies, a scarcity of diverse training environments, and the challenge of modeling complex attack physics. To address this, we introduce the **Parametric Graph Grammar for Procedural Content Generation (PGG-PCG)**, a novel stochastic generative model designed to reconstruct realistic network structures. Our approach leverages a parametric L-System formalism, $G = (V, \Sigma, R, S)$, where the device type alphabet Σ and probabilistic production rules R are empirically derived from real-world network data (LANL dataset). This architecture is implemented within a Stratified Generative Pipeline (SGP) that ensures topological validity (Phase I) by synthesizing realistic, structurally valid enterprise topologies (e.g., Fat-Tree, Spine-Leaf) using device adjacency probabilities, thereby avoiding the common issue of structurally invalid samples in pure generative models. We further ensure semantic integrity (Phase II) by injecting device capabilities via Stratified Sampling, and critically, guarantee adversarial realism (Phase III) by integrating the **Technique Inference Engine (TIE) as the state-transition oracle**. This delegates the complex, non-Markovian attack physics—a major limitation for standard DRL approaches—to a CTI validated probabilistic recommender (NDCG@20 = 0.18 ± 0.02). The PGG-PCG framework successfully produces inexhaustible, topologically, and semantically valid training environments, which is the necessary foundation for developing generalizable and robust DRL agents capable of autonomous network defense.

Introduction

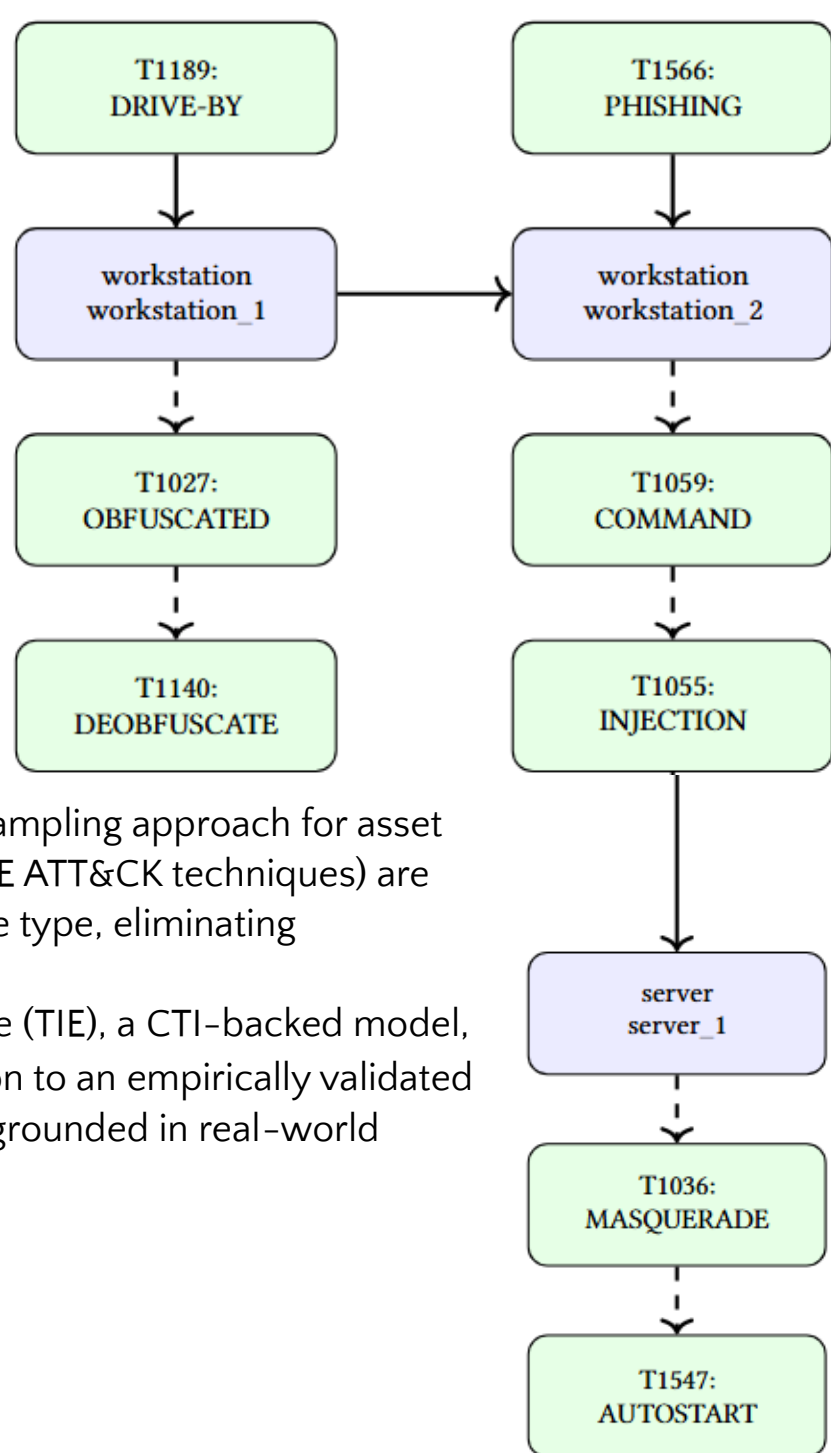
The development of autonomous cyber defense agents is severely limited by data scarcity and poor policy generalization. Current **Deep Reinforcement Learning (DRL)** environments rely on small, static, manually curated network graphs. Consequently, trained agents consistently overfit, demonstrating **brittle performance** when faced with **novel network configurations or unseen attack vectors**. We hypothesize that **Procedural Content Generation (PCG)**—typically a game development technique—can bridge this generalization gap by algorithmically constructing an unbounded stream of unique training environments.

To this end, we introduce the **Stratified Generative Pipeline (SGP)**, which reframes environment generation from static data modeling to dynamic content synthesis. As illustrated in the pipeline's workflow (Figure 2), the SGP first establishes semantic validity via data-driven structural priors.

Our contributions are threefold, addressing the core requirements for training generalizable cyber defense agents:

- Topological Fidelity:** We establish that Probabilistic Graph Grammars, conditioned on empirical metrics (Activity Volume, Protocol Profile, Centrality Score), successfully generate synthetic network topologies that accurately reflect known enterprise structures (e.g., Fat-Tree) rather than the non-representative scale-free or Erdős-Rényi models.
- Semantic Validity:** We design and implement a novel Stratified Sampling approach for asset injection, guaranteeing that all generated node capabilities (MITRE ATT&CK techniques) are semantically valid and compatible with the node's inferred device type, eliminating "hallucinated" attack actions.
- Adversarial Realism:** We integrate the Technique Inference Engine (TIE), a CTI-backed model, as the state transition oracle T . This delegates the $P(s|s', a)$ function to an empirically validated model, ensuring that attack path sequences are probabilistically grounded in real-world adversarial tradecraft.

Figure 1: Example MITRE technique transitions across and within devices



Methodology

The **Stratified Generative Pipeline (SGP)** fundamentally reframes Deep Reinforcement Learning (DRL) environment construction from static data modeling to dynamic, unbounded procedural content generation. Designed to synthesize JSON-serializable graphs compatible with standard GNN-based DRL observation spaces, the pipeline operates through comprehensive preparatory and dynamic generation phases.

Adversarial Synthesis & Data-Driven Priors

Before topology generation begins, the SGP constructs a robust action grammar by fusing static rule-based frameworks with dynamic Large Language Model (LLM) synthesis. The base grammar systematically maps the MITRE ATT&CK framework into five distinct roles: entrance, goal, key, task, and non-terminal task. To dynamically expand this mission space, the pipeline employs an LLM (Gemini 2.5 Flash) to conduct a conceptual inversion process on defensive mitigations, synthesizing novel adversarial bypass techniques classified as "key" capabilities.

Simultaneously, the pipeline establishes empirical structural priors using the Unified Host and Network Dataset (LANL). Using a **Statistical PCG (SPCG)** methodology, the framework ingests node-specific behavioral metrics—event activity volume, network protocol profiles, and topological centrality scores. These metrics softly bias the probability distributions for assigning device roles (e.g., domain controllers, workstations, IoT devices), successfully scaling environments to thousands of nodes while enforcing strict, enterprise-scale global distribution targets (e.g., 62% workstations, 8% network switches).

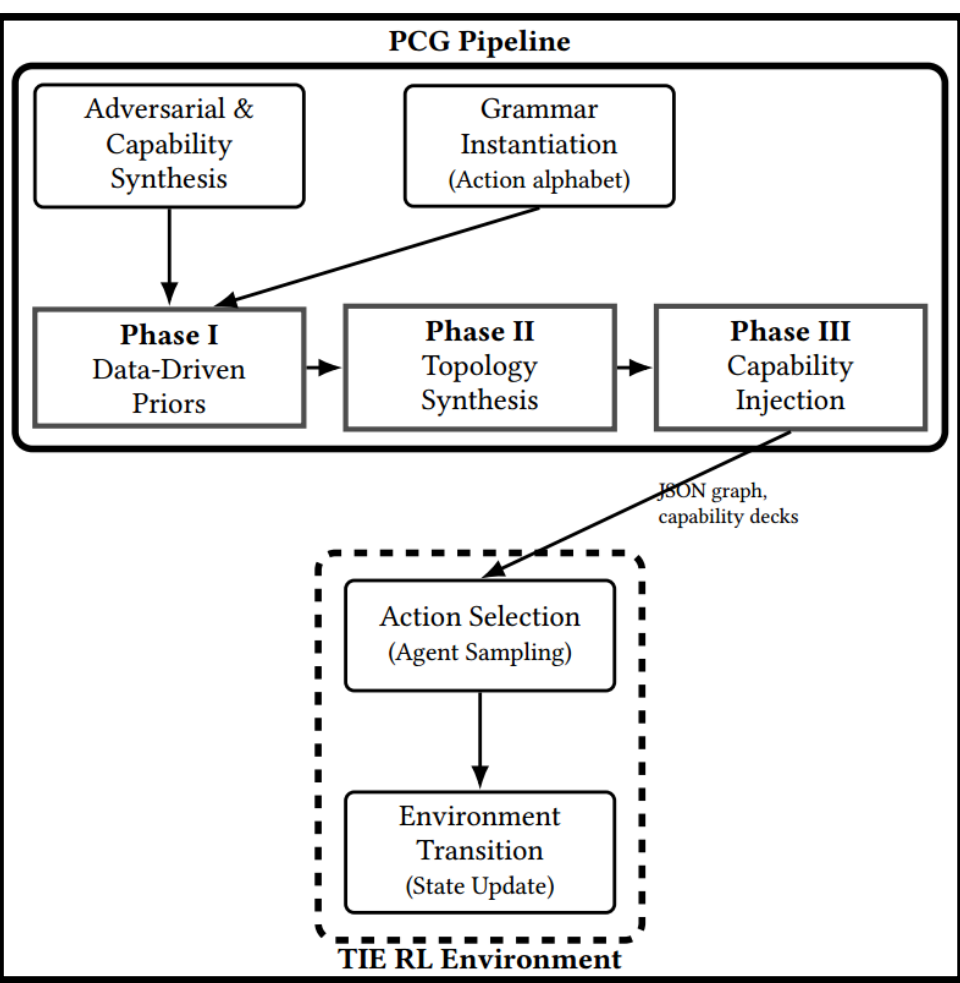


Figure 3: SGP and TIE Validation Workflow

pulls from device-specific capability decks bounded by a strict overlap threshold. The pipeline enforces two critical capability enrichment guarantees: each node is provisioned with a dense baseline minimum of benign NPC (Non-Player Character) actions, alongside randomly injected, contextually appropriate MITRE technique actions. This procedural injection eliminates "hallucinated" or structurally impossible attack vectors from the state space.

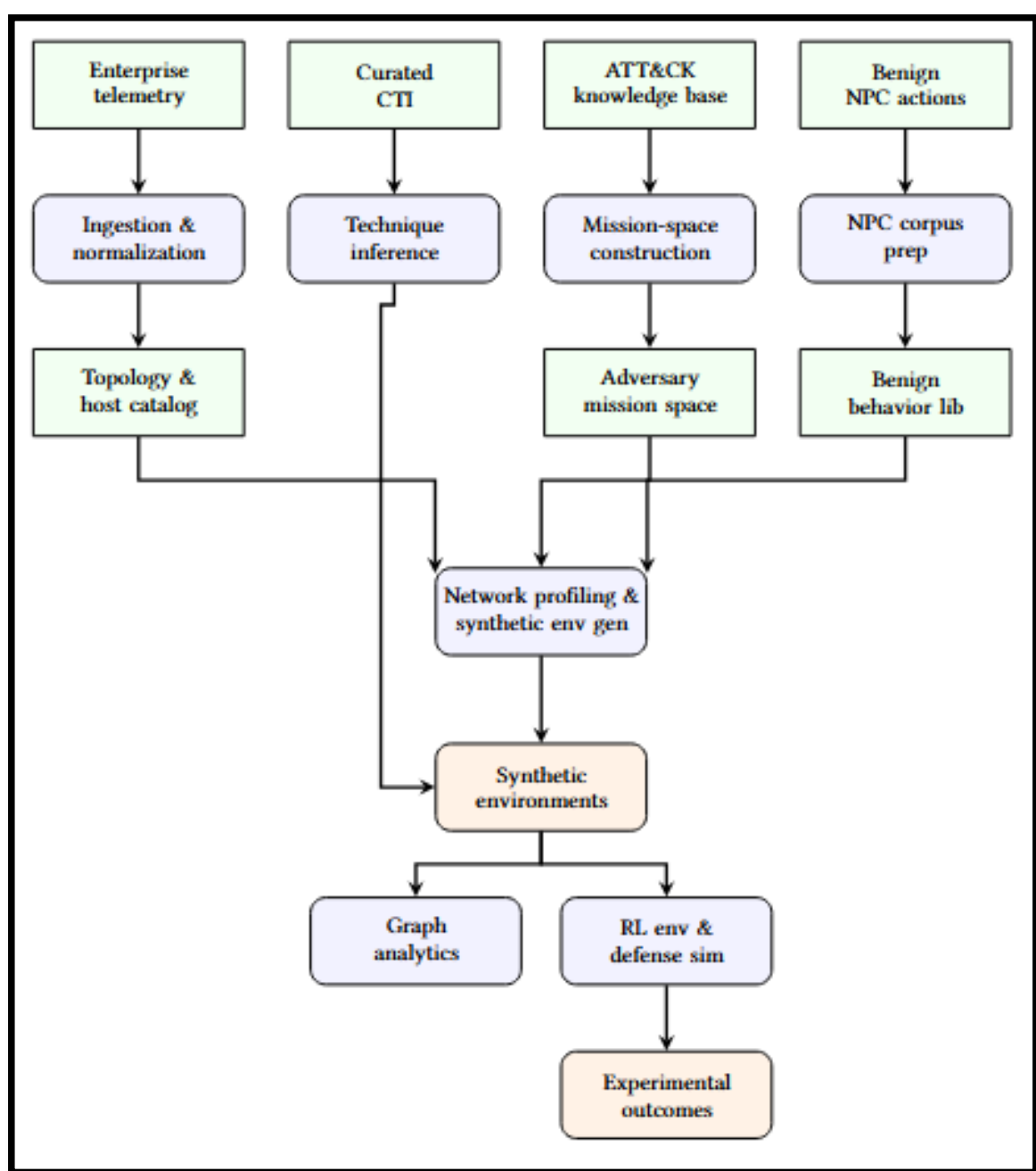


Figure 2: High-level experimental pipeline including telemetry ingestion and RL evaluation

Phases I & II: Parametric Topology Synthesis

To overcome the limitations of scale-free or Erdős-Rényi models, SGP implements a **Parametric Graph Grammar**. Unlike traditional string-based L-Systems, this generative model operates directly on graph matrices, using real-world adjacency statistics and degree distributions. The grammar is directed by parameterized Scenario Profiles (e.g., Corporate Office, Research Campus), which dynamically manipulate node branching factors, type weights, and structural templates. This algorithmically guarantees the synthesis of structurally valid, hierarchical architectures common to enterprise settings, such as Spine-Leaf or Fat-Tree topologies.

Phase III: Procedural Capability Injection

Following structural synthesis, SGP employs a rigorous Stratified Sampling protocol to inject semantically valid capabilities into the network nodes. To ensure each device features a distinct and highly specific attack surface, a parallelized process pulls from device-specific capability decks bounded by a strict overlap threshold. The pipeline enforces two critical capability enrichment guarantees: each node is provisioned with a dense baseline minimum of benign NPC (Non-Player Character) actions, alongside randomly injected, contextually appropriate MITRE technique actions. This procedural injection eliminates "hallucinated" or structurally impossible attack vectors from the state space.

Transition Oracle: The Technique Inference Engine (TIE)

Finally, the generated environment avoids brittle, hand-crafted transition logic by integrating an enhanced MITRE Technique Inference Engine (PyTIE) directly into the RL loop as a state-transition oracle. Delegating the non-Markovian $P(s|s', a)$ transition function to this oracle ensures that environment dynamics mimic real-world adversarial tradecraft. Utilizing embedding-based models optimized via hyperparameters (e.g., Implicit BPR and WALS recommenders), PyTIE was trained on approximately 44,000 empirical observations distilled from 6,000 curated Cyber probabilistically evaluates its success likelihood based on the adversary's execution history, creating a truly hallucination-resistant training simulation.

Results

Topological Fidelity and Scale – Analysis of the generated network reveals a structurally sound, non-scale-free architecture indicative of an enterprise-level Fat-Tree or Spine-Leaf topology. The Stratified Generative Pipeline successfully synthesized and profiled 13,599 individual devices without any mismatches between assigned action capabilities and device type compatibility.

Device	Type	Prob.
workstation	62.0%	N/A
domain_controller	0.5%	1:200
shared_server	2.0%	1:50
printer	2.0%	1:50
router	1.0%	1:100
firewall	1.0%	1:100
network_switch	8.0%	1:12
vpn_gateway	0.5%	1:200
wireless_access_point	10.0%	1:10
storage_array	0.5%	1:200
backup_server	0.5%	1:200
security_appliance	1.0%	1:100
iot_device	10.0%	1:10
legacy_system	0.5%	1:200
test_lab_machine	0.5%	1:200
Total	100.00%	

Table 1: Target Distribution of Device Types in an Enterprise Environment

Table 2 demonstrates the framework's ability to strictly adhere to enterprise target distributions. The procedural generation correctly populated the environment with a baseline of 8,272 workstations, while accurately concentrating critical backbone infrastructure. For example, domain controllers and routers successfully met their exact global distribution targets (1:200 and 1:100 ratios, respectively), proving the efficacy of the metric-aware bias algorithm. Furthermore, capability density was logically distributed; high-activity nodes (e.g., workstations, shared servers) received rich action sets, while specialized hardware (e.g., backup servers) received highly targeted capabilities.

Structural Adjacency and Aggregation – The semantic and structural validity of the generated mission space is visually corroborated by the ego community networks. **Figure 3** illustrates the simulated topography surrounding a central "Active.Directory" device, with nodes colored by their respective device types and directional edges mapping their interactions.

The adjacency profiles explicitly validate the hierarchical engineering imposed by the parametric graph grammars. The visualization demonstrates that endpoint devices like workstations, IoT devices, and printers

cluster as expected, connecting securely into infrastructure tiers. Wireless access points and network switches effectively operate as primary aggregation hubs, exhibiting high connectivity probabilities to endpoint clusters, which perfectly mirrors real-world operational environments.

Adversarial Logic Validation – To ensure the Deep Reinforcement Learning (DRL) agent faces realistic attack physics, the environment relies on the **PyTIE** transition oracle, which was optimized via automated hyperparameter sweeps across various recommender algorithms.

Model	MSE	Precision	Recall	NDCG
TopItemsRecommender	0.327	0.0025	0.0317	0.0176
FactorizationRecommender	8.406	0.0015	0.0178	0.0105
BPRRecommender	2.527	0.0073	0.1	0.0507
ImplicitBPRRecommender	0.78	0.0121	0.1661	0.0922
ImplicitWALSRecommender	0.508	0.0083	0.1103	0.0591

Table 3: Performance of Recommender Models in the PyTIE

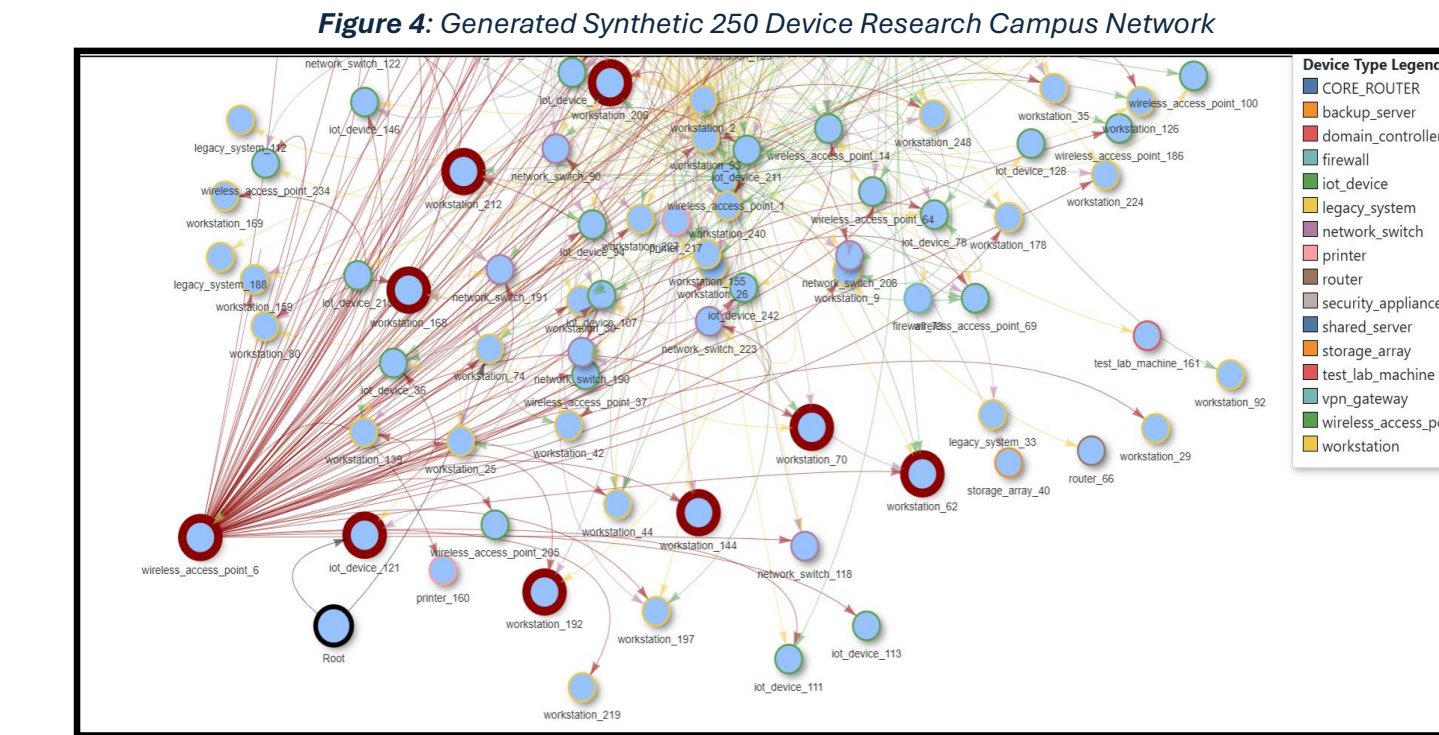
Additionally, the integrated Implicit BPR Recommender demonstrated a substantial performance surge (0.0922 NDCG) compared to legacy implementations. By contrast, baseline algorithms like the Factorization and TopItems recommenders posted high error rates and the lowest ranking quality (0.0105 NDCG for Factorization), confirming their inability to model complex cyber threat intelligence (CTI) sequences.

Conclusion & Future Direction

The **Stratified Generative Pipeline (SGP)** fundamentally solves the critical generalization failures of **Deep Reinforcement Learning (DRL)** cyber defense agents by algorithmically synthesizing unbounded, high-fidelity training environments. By shifting from static data modeling to dynamic content synthesis, SGP ensures agents are trained in environments possessing true adversarial richness.

Key Takeaways:

Topological Realism: As visualized in the **Ego Community Graph of a Campus Router (Figure 4)**, our Parametric Graph Grammar successfully generates hierarchical, enterprise-scale topologies (e.g., Fat-Tree, Spine-Leaf). The visual clearly demonstrates realistic clustering, endpoint devices like workstations and IoT seamlessly connect into primary aggregation hubs like share servers and access points, accurately mirroring operational realities rather than simplistic, non-representative scale-free models. The red circles represent the devices tagged with MITRE techniques predicted from the TIE model and embedded onto a valid lateral network path.



Counterfactual Scenario Generation for Zero-Day Defense: Currently, the SGP relies on historical observational priors, which limits exposure to rare, "what-if" edge cases. We will extend Phase II into a Counterfactual Generative Pipeline by redefining the Parametric Graph Grammar as a **Deep Structural Causal Model (DSCM)**. This will generate deterministic counterfactual training environments through a three-step causal inference process:

Step Index	Node ID	Technique ID	Technique Name	Role
0	Root			prelude
1	iot_device_121	T1046	Network Service Scanning	origin
2	workstation_49	T1591.004	Gather Victim Org Information: Business Relationships	hop
3	workstation_206	T1003.007	OS Credential Dumping: Proc Filesystem	hop
4	workstation_62	T1562.002	Impair Defenses: Disable Windows Event Logging	hop
5	workstation_144	T1055	Process Injection	hop

Table 4: Excerpt of Attack Path of Synthetic 50 Device Research Network

- Abduction:** Utilizing Normalizing Causal Flows (NCF) to capture specific background conditions (hidden "noise" variables) compatible with an observed network graph.
- Action:** Applying manual interventions to simulate targeted, rare edge cases (e.g., forcing a firewall bypass or zero-day compromise).
- Prediction:** Combining the unique hidden states with the modified rules to generate logically consistent, counterfactual network topologies that reflect the global impact of the attack.

Enhancing the Technique Inference Engine (TIE): To overcome the "cold start" problem for new techniques and better model complex, non-linear attack paths, the TIE will transition away from linear matrix factorization models. Future iterations will explore advanced architectures such as **Graph Neural Networks (GNNs)**, **Deep Factorization Machines (DeepFM)**, **Neural Collaborative Filtering (NCF)**, and **Transformer-based Sequence Models**.

Immediate Integration & Evaluation: Near-term efforts will resolve over 5,300 unmapped "orphan" network components and finalize the integration of SGP datasets into the primary DRL environment wrapper. Ultimate success will be quantified by measuring the trained agents' policy transferability scores against diverse, unseen static network configurations.

References

