C-BYTES

Research Showcase and Student Competition

Guidance and Q&A Session

November 4, 2026

College of Information Science & Technology

University of Nebraska Omaha

C-BYTES Resources

- Main C-BYTES Page:

  https://www.unomaha.edu/college-of-information-science-and-technology/research-labs/resources/cbytes.php

- C-BYTES Resources and Guidelines page:

  https://www.unomaha.edu/college-of-information-science-and-technology/research-labs/resources/cbytes_resources.php

- 3 Minute Thesis (3MT) previous winners (video):

  https://threeminutethesis.uq.edu.au/watch-3mt

C-BYTES 2026 Official Rules

COLLEGE OF INFORMATION SCIENCE & TECHNOLOGY

C-BYTES Research Showcase and Student Competition

Official Rules

While Quick-Share and Poster-BYTES challenges are completely different in their execution, both need visual elements to enhance understanding within the framework and limitations given. Visuals must be clear, relevant, and well-organized, contributing to the overall communication of the project.

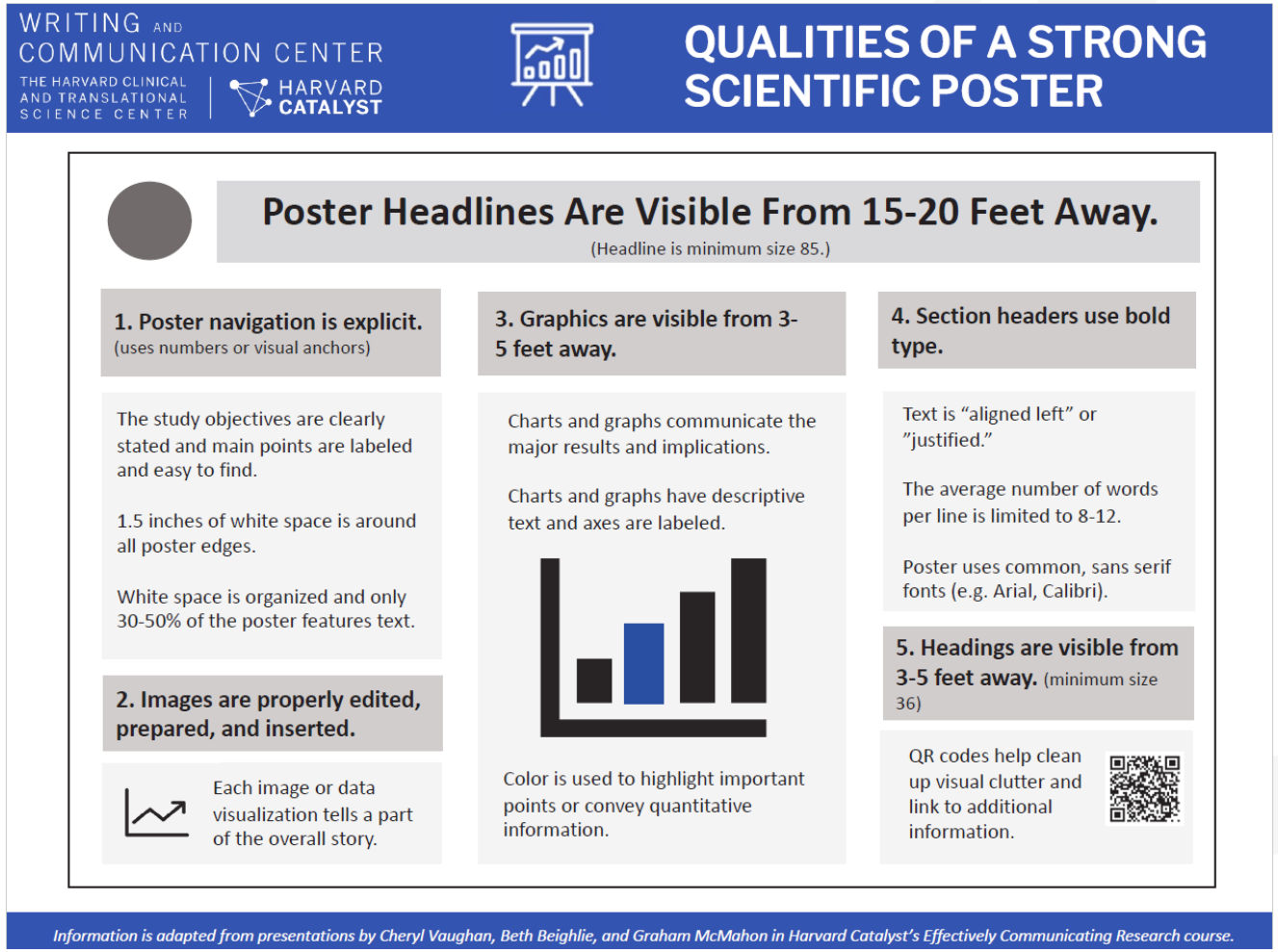
Code of Conduct: All participants are expected to uphold the University of Nebraska at Omaha Student Code of Conduct and maintain professionalism throughout the event.

Poster-BYTES Tips

- Create a well-designed research poster
- Prepare a verbal “pitch”
- Prepare to have the “conversation” with your audience.

https://catalyst.harvard.edu/writing-communication-center/visualize-science/posters/

Poster-BYTES: Poster Design



Poster-BYTES: Poster Design

High Precision Congestion Control (HPCC)

Simulation Refactoring

College of Information Science & Technology, Department of Computer Science

CONTRIBUTORS:

- Daniel Kaseya
- Xiaoqian (Tiffany) Zhang

Abstract

Continuous of the NS-3 network simulator, transitioning from "waf" build system to CMake has created a reproducibility challenge for projects built using the deprecated system. One notable example of is The High Precision Congestion Control (HPCC) project, a notable contribution to data center networking. This project aims to refactor the simulation of the HPCC from the legacy version it was built on to the most updated and stable version of NS-3. Additionally, the refactoring will be validated through a series of tests for validation and correctness .

Introduction

- Data center network link speed has increased from 1 Gbps to 100 Gbps.
- Ultra-low latency and high bandwidth are crucial for modern and future high-speed networks.
- Large-scale Remote Direct Memory Access (RDMA) networks.
- High speed leads to congestion and deep packet queueing, impacting latency-sensitive flow and congestion Handling.
- Current solutions (RDMA) faces challenges reconciling low latency, high bandwidth utilization, and high stability.

Materials

- **Simulation Environment:** network simulator 3 (ns-3).
- **Hardware/Software Setup:** CLion as IDE.



Figure 1: Network Simulator

Methodology

- Simultaneously achieves ultra-low latency, high bandwidth utilization, and network stability.
- Use precise congestion control with in-Network Telemetry (INT).
- **In Network Telemetry (INT):** A monitoring technology that allows switches and routers to embed real-time information (txBytes, qLen, timestamps, etc…) about network conditions in data packets.

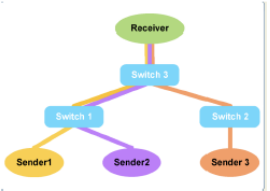


Figure 2: HPCC workflow

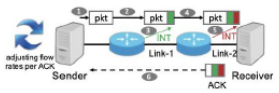


Figure 3: HPCC Framework

Anticipated Results

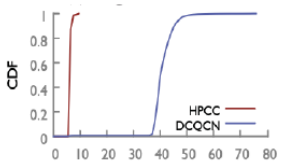


Figure 4: Elephant-Mice Latency (DCQCN vs HPCC)

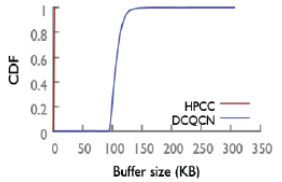
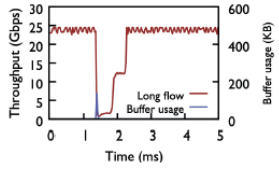


Figure 5: NS-3 – Elephant – Mice Queue Size (DCQCN vs HPCC)



(c) *Incast* HPCC

Figure 6: Incast HPCC: Long flow and Buffer usage

Plan

- **Refactoring & Migration:** Refactor all simulation parameters into a dedicated class for improved readability and migrate modified libraries and headers to align with the modern CMake build framework.
- **Validation & Testing:** Conduct a series of testbeds to accurately mimic the simulation environment and validate the initial findings.

Acknowledgements

Sincere gratitude is extended to Professor Xiaoqian (Tiffany) Zhang, PhD, for her invaluable guidance and support throughout this initial process of this project.

University of Nebraska Omaha

Opening the Black Box: Mechanistic Interpretability

CONTRIBUTORS: Anirban Pal

College of Information Science & Technology

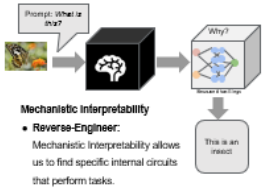
What is MI

Mechanistic Interpretability (MI) aims to reverse-engineer how neural networks (LLMs and VLMs) process information and make decisions by identifying the specific internal circuits and components responsible for learned behaviors. This approach seeks causal insights into model computation, unlike post-hoc methods. Advances leverage tools like **circuit tracing, sparse autoencoders (SAEs), activation patching**, and **logit attribution**.

Why It Matters?



- **Trust & Safety:**
  For AI in high-stakes applications (medicine, autonomous systems, finance), we need to ensure:
  - **Fairness:** Is it biased?
  - **Accountability:** Who is responsible when errors occur?
  - **Reliability:** Can we trust its decisions?
- **Scientific Understanding:**
  Move beyond observing behavior to understanding the mechanisms that drive it.
  Example: *Why did the model correctly identify a butterfly as an "insect"? What internal features or circuits led to that specific classification?*
- **Targeted Debugging:**
  Pinpoint exact internal causes of errors (e.g., hallucinations, adversarial vulnerabilities) for precise fixes, rather than trial-and-error.
- **Inspiration for Better AI:**
  Discover efficient or novel mechanisms within models that can inform the design of future, more robust, and interpretable architectures.



Mechanistic Interpretability

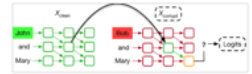
- **Reverse-Engineer:**
  Mechanistic Interpretability allows us to find specific internal circuits that perform tasks.
- **Debugging & Control:**
  Identify and fix internal failures (*e.g., reduce bias, prevent hallucinations, improve robustness*).
- **Build Safer AI:**
  By understanding internal "wires" (circuits), we can better control, align, and build more trustworthy and ethical AI systems.

Key Mechanistic Interpretability Techniques

Sparse Autoencoders (SAE)
SAEs help reveal human-interpretable features inside large neural networks by learning a sparse set of latent units from intermediate activations.
When a model's internal neurons are polysemantic (language model encoding "red," "round," and "toy" at the same time), SAEs act as feature separators. Trained with an L1 sparsity penalty, they learn an overcomplete dictionary where each latent ideally corresponds to one concept.
For instance, given the activation produced by an image of a red ball, an SAE may separate it into distinct latent units such as "redness," "roundness," and "object-toy." These disentangled latents can then be individually probed or manipulated, allowing researchers to test causal hypotheses (e.g., amplifying only the "redness" feature) and making the model's internal representations more transparent and mechanistically interpretable.

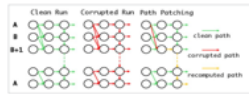
Activation Patching
Precise causal pathways of information flow through the entire network.
X_{clean} : A clean prompt, "The Eiffel Tower is in the city of" with an associated answer, "Paris".
$X_{corrupt}$: A corrupted prompt "The colosseum is in the city of".

1. Clean Run: Test on X_{clean} and cache activation of a set of given model components.
2. Corrupted Run: Test on $X_{corrupt}$ and record the model outputs.
3. Patched Run: Test on $X_{corrupt}$ with a specific model component's activation restored from the cached value of the clean run.



Path Patching
Identifies the precise causal role of specific information pathways (edges) between components. Path patching includes three times of forward pass.

1. Clean Run: Test on X_{clean} and cache the value signal of a specific edge (e.g., the output contribution of **Attention Head** *h* at the token "Eiffel" → flowing into the **residual stream**).
2. Corrupted Run: Test on $X_{corrupt}$ and record the model outputs (likely resulting in "Rome" or another non-Paris city, or a less confident prediction).
3. Patched Run: Test on $X_{corrupt}$ with **only the cached signal from the specific clean edge** (e.g., the value contribution from **Attention Head** *h* restored into the corresponding edge/path in the corrupt run. All other internal components and pathways remain from the corrupted run.



Key Insights

Internal Circuits Are Real (But Complex): MI has empirically shown that "black box" neural networks implement learned behaviors using identifiable, specific internal circuits—not just diffuse, uninterpretable activity.

Causality Is King: The field's power lies in its causal interventionist approach to move beyond correlations and pinpoint *why* models behave a certain way.

VLMs Introduce New Hurdles: Cross-modal architectures bring unique challenges requiring specialized MI adaptations.

Transparency Doesn't Scale Naturally: Simply building bigger models doesn't yield more interpretable ones; increased scale often brings increased complexity.

Bridging Engineering & Science: MI advances our understanding of AI systems, similar to how neuroscience studies the brain, offering deep insights for future AI development.

Open Challenges

1. **Generalization:** Findings are often task/model-specific; circuits don't reliably generalize across different architectures or tasks.
2. **Complexity & Affordability:** Techniques are computationally expensive (e.g., patching large models) and require significant expertise, limiting broader accessibility.
3. **Polysemanticity & Redundancy:** Units encoding multiple concepts; multiple components for the same role.
4. **Modality Dominance (VLMs):** Textual priors often override visual evidence, leading to attenuated visual influence.
5. **Scale Limitation:** Interpretability doesn't scale linearly with model size; complexity increases.
6. **Encoder Fidelity:** Insufficient visual encoding can mask fine-grained mechanisms in VLMs.

Acknowledgements

Supported by the College of Information Science & Technology, University of Nebraska Omaha.
Presented at UNO C-BYTES Research Showcase 2025.
Guided by Dr. Ann Fruhling and Dr. Xin Zhong

References

1. Wang et al., 2022
2. Shu et al., 2025
3. Joseph et al., 2025
4. Golgoon et al., 2024
5. Zhang et al., 2025
6. Bahador et al., 2025
7. Liu et al., 2025
8. Olah et al., 2017
9. Vaswani et al., 2014

Nebraska Omaha

Poster-BYTES Presentation: Pitch

- Open with your **Name** and **Role**
- What motivates the research? State the **Problem** and **Why it Matters**.
- Give *just enough* background: what's **Known**, the **Gap**, and your **Hypothesis/Approach**.
- Share the "news": your **Key Result** or Insight and what **Impact** that has.
- Land the takeaway: **What's Next** (next steps, collaboration, or specific support).

Delivery:

- Clear voice, steady pace, eye contact, purposeful gestures—you're narrating your research story.
- Keep it to **45–60s** ready to expand if asked.

https://catalyst.harvard.edu/writing-communication-center/visualize-science/posters/

Poster-BYTES Presentation: Conversation

- Read the room: learn audience background and goals.
- Start warmly: smile, greet, introduce yourself, and ask them if they are familiar with your topic.
- Customize the depth for audience; **avoid jargon**; check for clarity as you go.
- Show, don't just tell: **point to figures**, **use simple analogies** or short stories.
- Welcome questions, pause to listen, and navigate to the relevant section.
- Pivot as needed: zoom out for big picture (irl **why does this matter?**) or dive deeper on methods/results.
- Close and connect with viewers: recap one takeaway and invite follow-up.
- Request viewers scan the competition QR Code and complete the assessment rubric.

https://catalyst.harvard.edu/writing-communication-center/visualize-science/posters/



Relating Answer Set Programming and Many-sorted Logics for Formal Verification

Zach Hansen

-PhD Computing and Information Science (in progress)

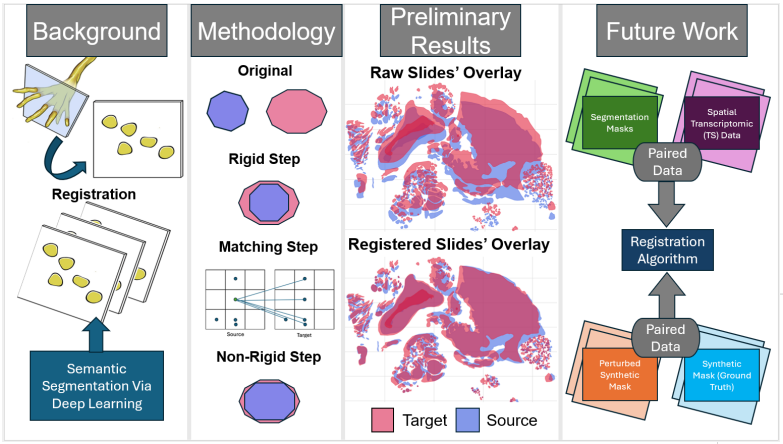
Advisor: Dr. Yuliya Lierler

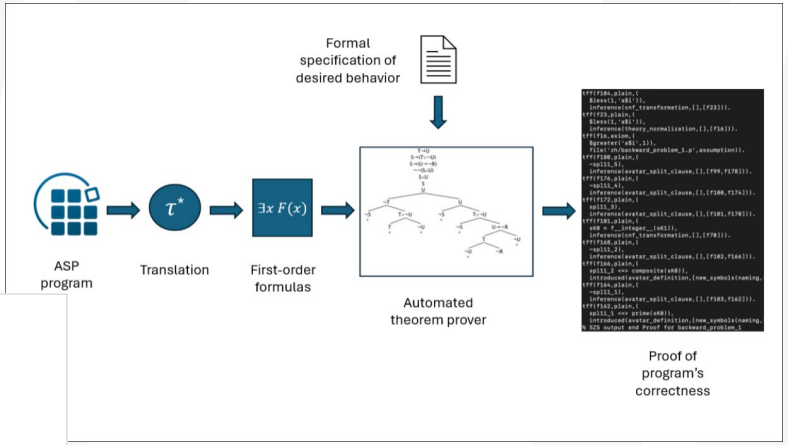
University of Nebraska Omaha

Quick-Share Presentation

- Your presentation should
  - Engage us.
  - Inform us.
  - Communicate to a **lay audience** – Why should **we** be excited?
  - Connect research to big picture – Why is this important **to the rest of us**?
  - Touch on **the most interesting** aspect about your research.
- Making an impactful slide
  - Start with the verbal message. Create a visual that clarifies it.
  - Don’t let it **be** your presentation. It is an **illustration to clarify** your talk.
  - It **should not distract** attention away from your presentation.

Quick-Share Slide (Examples)





The biological father of Sam is John.

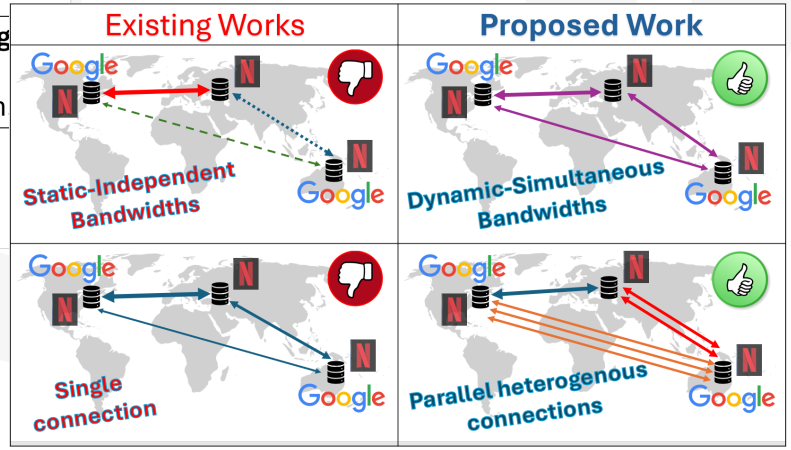
Predicate Encoding:

father(sam, john).

Function Encoding

father(sam) := john





Use 3 minutes wisely

- Don’t introduce yourself, don’t do acknowledgements, don’t show data.
- Introduce how your research relates to the bigger picture--why should we care?
- Carefully design your slide, don’t overcrowd it or use too much color.
- Only include most relevant things.
- Make sure images are good quality.
- **Use simple familiar analogies. Think of 3 to 5 words your audience will remember.**
- Use excited and engaged body language: smiling, gestures, movement, and eye-contact.
- Finish by bringing the audience back to the big picture.

Use up your time, but don’t go over it!

- Practice.
- Practice out loud.
- Practice standing up and performing the talk.
- Practice and get feedback from a variety of people.

https://threeminutethesis.uq.edu.au/resources/3mt-competitor-guide

Resources

- Examples of single slides from 3MT®
  - https://ecampusontario.pressbooks.pub/scientificcommunication/chapter/the-3-minute-thesis/
- Examples of research posters
  - https://www.unomaha.edu/college-of-information-science-and-technology/public-health-informatics-research-lab/posters/index.php
- View scoring/judging rubrics
  - https://www.unomaha.edu/college-of-information-science-and-technology/research-labs/collaboratoriums/cbytes.php

We will provide T-Shirts to wear during the competition. Not a requirement, but we hope you will!

Contact Information

Kelly L. Hughes

Proposal Development Manager
College of IS&T | PKI 222
University of Nebraska at Omaha | unomaha.edu

402.554.3383
kellyhughes@unomaha.edu



90 or TTY 402.554.2978 or the Accessibility Services Center (phone: 402.554.2872). UCTEMP20