Making the Self-Driving “Net” Work

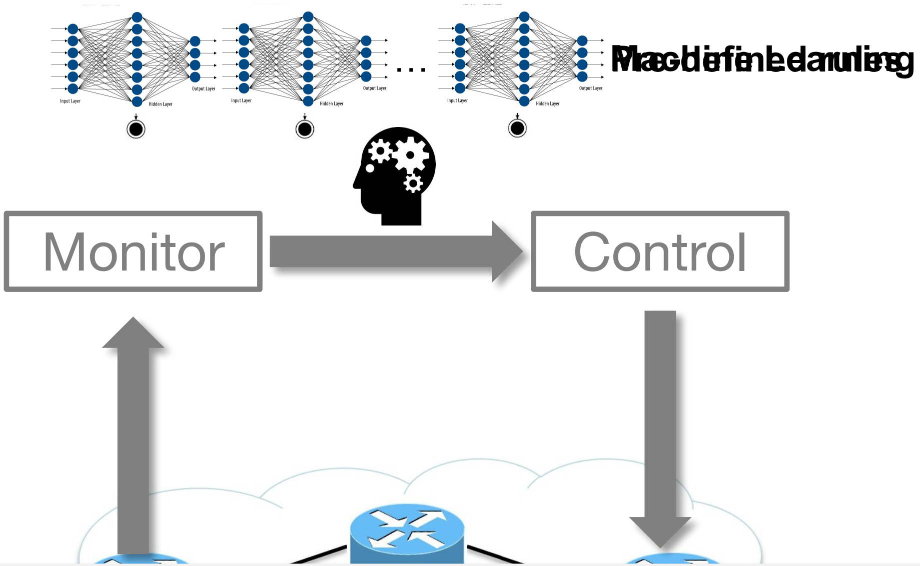
Developing Production-Ready ML Models for Self-Driving Networks

Arpit Gupta

Associate Professor, UC Santa Barbara

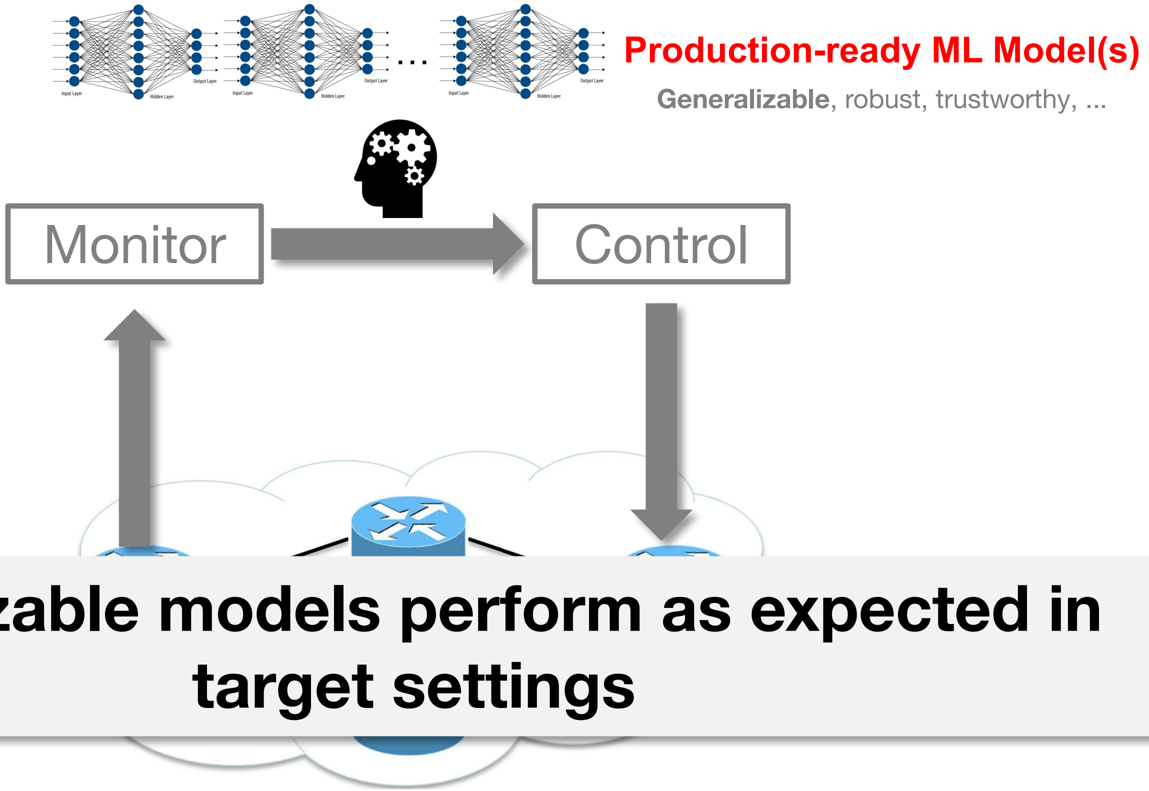
Faculty Scientist, Berkeley Lab (ESnet)

Self Driving Networks (AIOps)



Enable secure & performant connectivity with minimal human interventions

Key Requirement



Generalizable models perform as expected in target settings

AI/ML for Network Security: The Emperor has no Clothes

https://trusteeml.github.io/

Arthur S. Jacobs
UFRGS, Brazil
asjacobs@inf.ufrgs.br

Roman Beltiukov
UCSB, USA
rbeltiukov@ucsb.edu

Walter Willinger
NIKSUN Inc., USA
wwillinger@niksun.com

Ronaldo A. Ferreira
UFMS, Brazil
raf@facom.ufms.br

Arpit Gupta
UCSB, USA
arpitgupta@ucsb.edu

Lisandro Z. Granville
UFRGS, Brazil
granville@inf.ufrgs.br

Model Generalizability

Lack of generalizability attributable to at least 3 **underspecification issues**

- **Shortcut Learning**

  Model takes shortcuts to classify data (**cheating**)

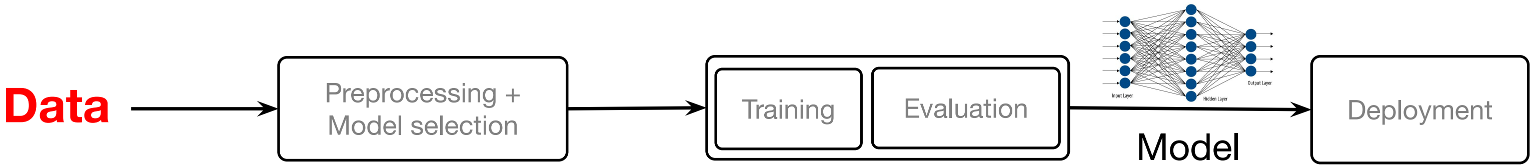
- **OOD Issues**

  Model does not generalize to out-of-distribution samples (**rote learning**)

ML models in networking are vulnerable to these underspecification issues!

Fundamental Roadblocks

Places the responsibility on users to find the **right data,**
i.e., data that enables developing **generalizable** ML model

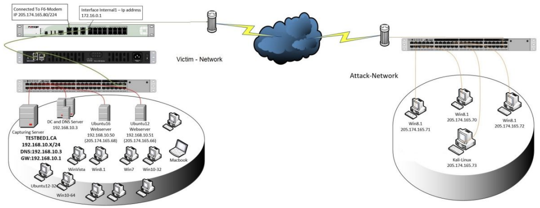


Standard ML Pipeline

For any given learning problem and target environment
what is the “right” data

Fundamental Roadblocks

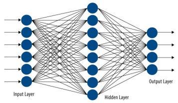
Networking problems necessitate **endogenous** data collection



Data Generation

↓

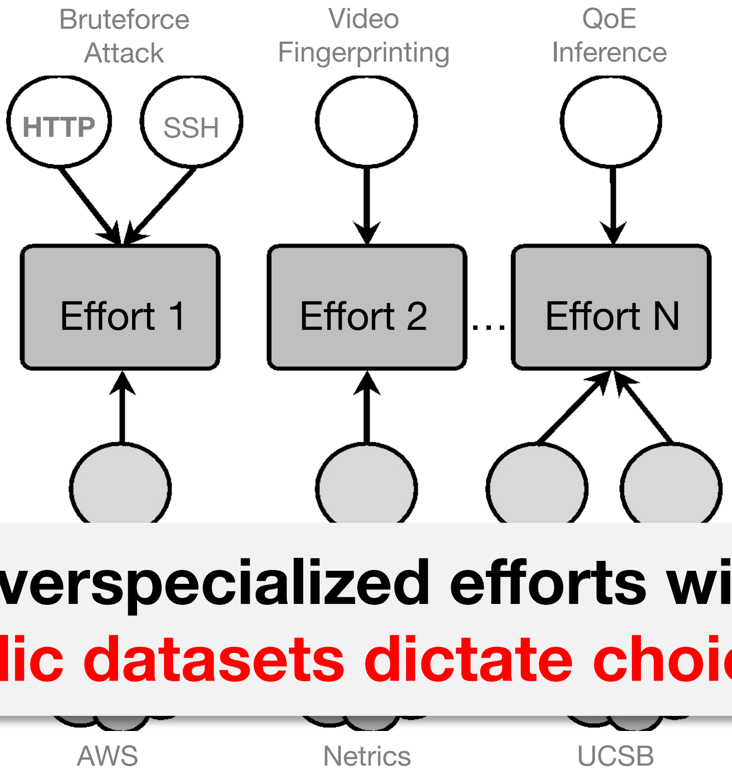
Data → Preprocessing + Model selection → Training | Evaluation → Model → Deployment



Standard ML Pipeline

For any given learning problem and target environment
how to generate the "right" data?

Data Generation Challenges



Learning Problems

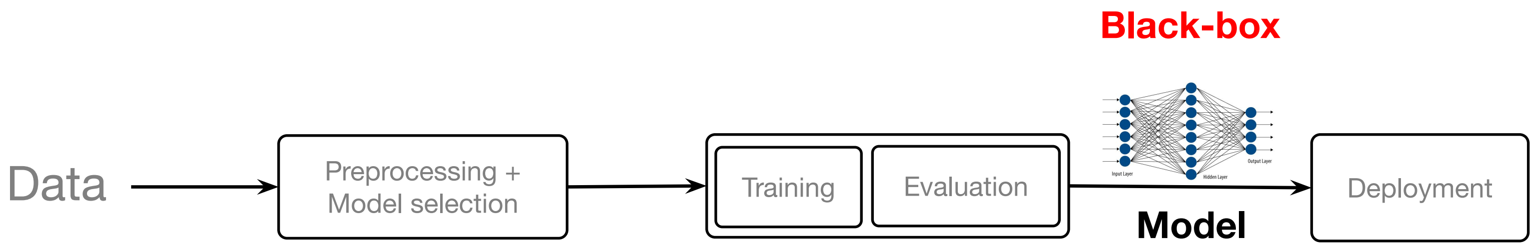
Fragmented **data generation**

Network environments

Overspecialized efforts with limited extensibility;
public datasets dictate choice of problems & models

Fundamental Roadblocks

Outputs the most performant model, with little to no insights into model's decision-making (**black-box**)

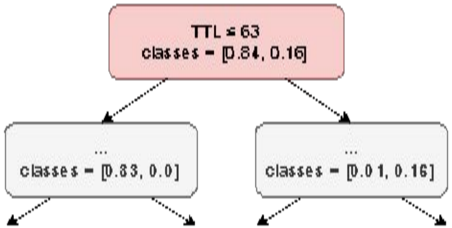


Standard ML Pipeline

How to assess if resulting models are generalizable, i.e., not underspecificied?

Example: HTTP Brute Force Attack Detection Problem

Learning shortcut

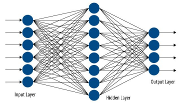


How to **generate better data** at scale?

Is this model **underspecified**?

F1-score: 0.99

CIC-IDS → Preprocessing + Model selection → Training | Evaluation →

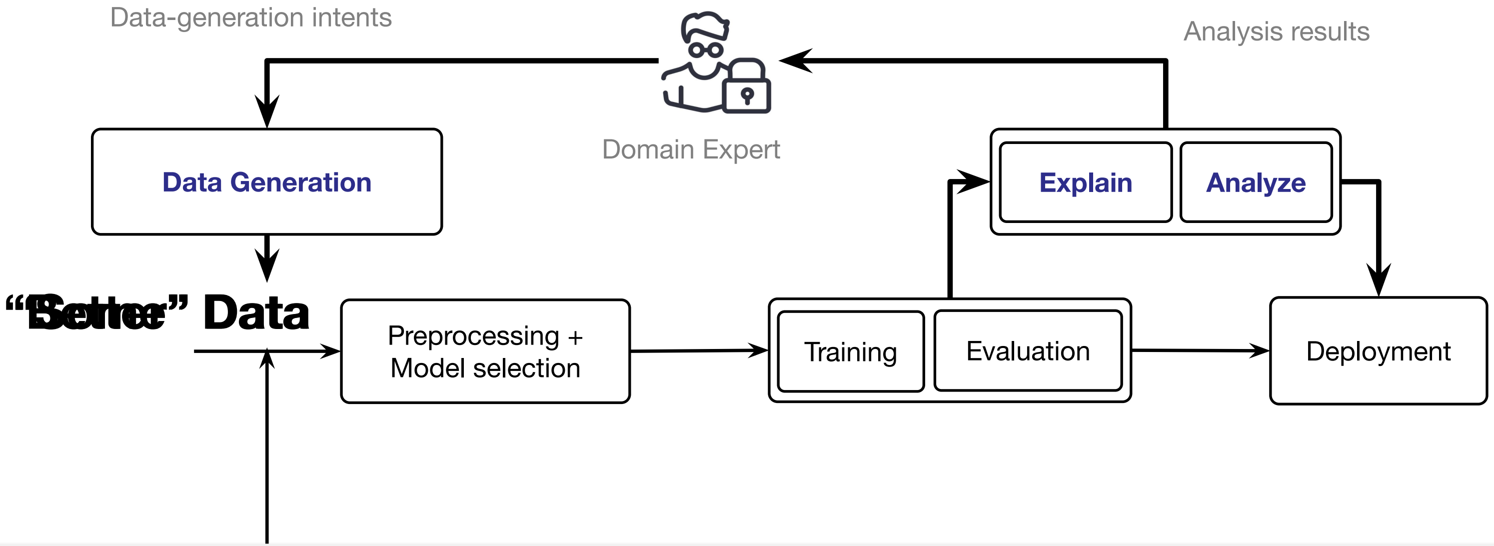


Is this the **right** data?

Standard ML Pipeline

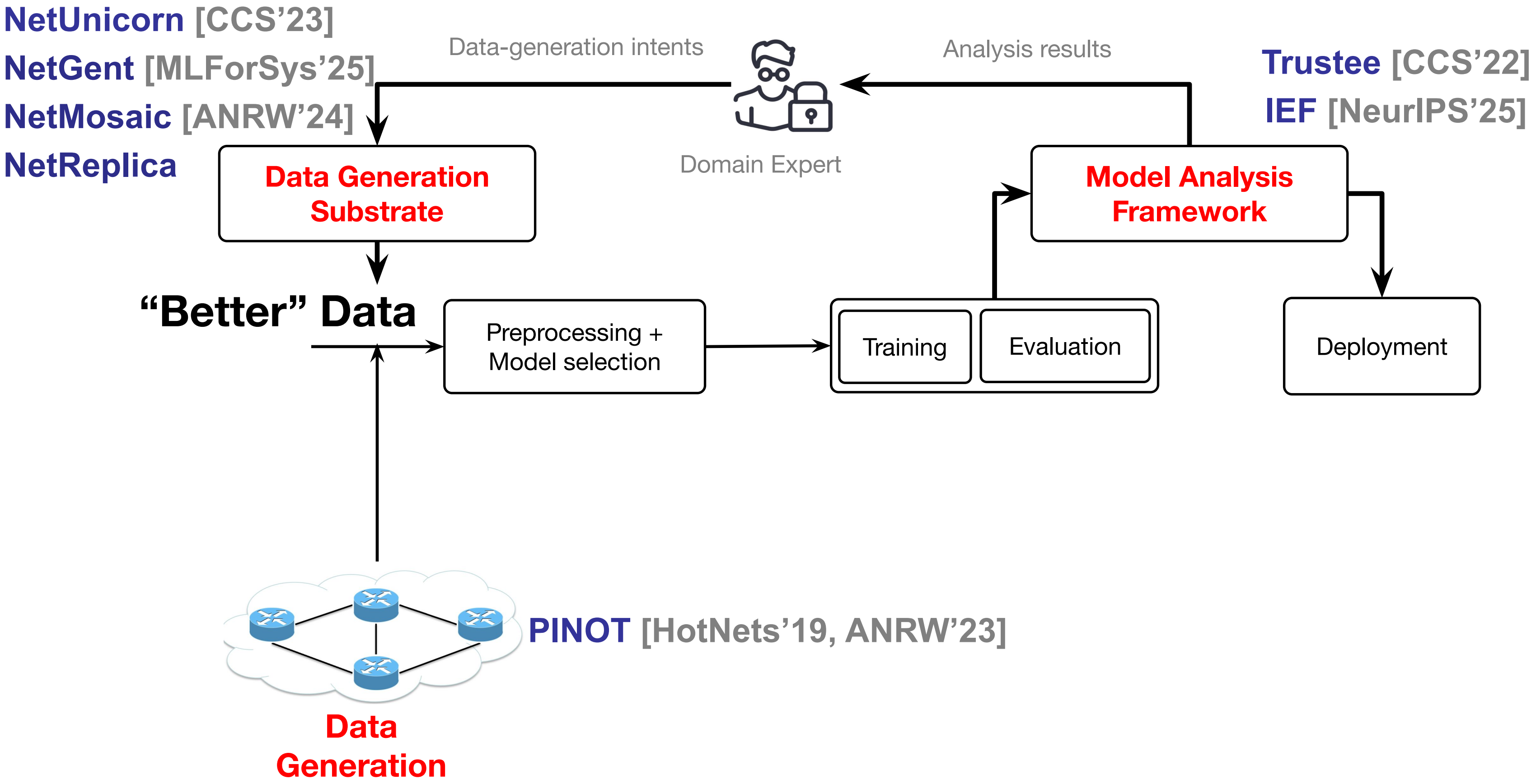
Answering these questions critical for developing generalizable ML artifacts for networking

Our Approach: Closed-Loop ML Pipeline



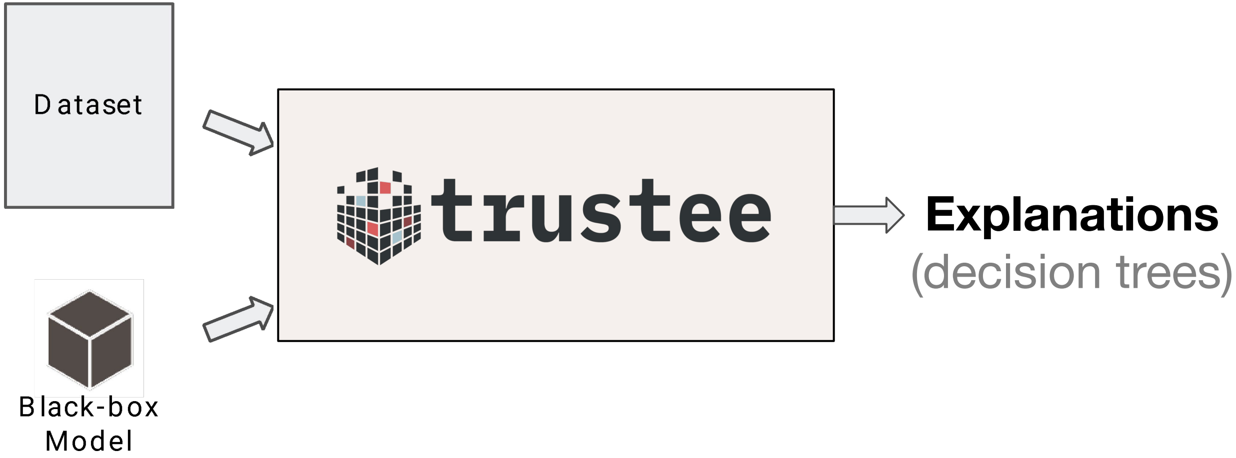
For any given learning problem and environment,
iteratively fix underspecification issues to generate “better” data

Realizing Closed-Loop ML Pipeline



Trustee

A **post-hoc global** model explainability framework



Model-agnostic, high-fidelity, low-complexity, and stable decision trees to explain model's decision-making

Generates a Trust Report

```
Classification Trust Report

Summary

Blackbox
    Model:              RandomForestClassifier
    Dataset size:       947072
    Train/Test Split:   70.00% / 30.00%

    # Input features:   61
    # Output classes:   5

Whitebox
    Explanation method: Trustee
    Model:              DecisionTreeClassifier
    Iterations:         1
    Sample size:        50.00%

    Decision Tree Info
    Size:               2437
    Depth:              31
    Leaves:             1219
    # Input features:   18 (29.51%)
    # Output classes:   5 (100.00%)

Top-k Whitebox
    Explanation method: Trustee
    Model:              DecisionTreeClassifier
    Iterations:         1
    Sample size:        50.00%

    Decision Tree Info
    Size:               9
    Depth:              4
    Leaves:             5
    Top-k:              1
    # Input features:   -
    # Output classes:   5 (100.00%)

Performance
              precision    recall  f1-score   support

           0      1.000     0.912     0.954     24408
           1      0.752     0.910     0.824      1872
           2      0.929     0.827     0.875     10994
           3      0.997     0.929     0.962     65188
           4      0.958     0.997     0.978    181660

    accuracy                          0.967    284122
   macro avg      0.927     0.915     0.918    284122

Fidelity
              precision    recall  f1-score   support

           0      1.000     1.000     1.000     22254
           1      1.000     1.000     1.000      2265
           2      0.969     0.965     0.967      9781
           3      0.998     0.998     0.998     60768
           4      0.998     0.998     0.998    189054

    accuracy                          0.997    284122
   macro avg      0.993     0.992     0.993    284122

Fidelity
              precision    recall  f1-score   support

           0      0.000     0.000     0.000     22254
           1      0.000     0.000     0.000      2265
           2      0.000     0.000     0.000      9781
           3      0.544     0.957     0.694     60768
           4      0.875     0.821     0.847    189054

    accuracy                          0.751    284122
   macro avg      0.284     0.356     0.308    284122

| Feature      | # of Nodes (%) | Data Split % - ↓ |
| TCP Src Port | 700 (57.47%)   | 782029 (37.11%)  |
```

Lowers the threshold of detecting underspecification issues

Trustee in Action

Problem	Dataset(s)	Model(s)	Underspecification Issues
Detect VPN traffic	Public VPN dataset [20]	1-D CNN [61]	Shortcut learning
Detect Heartbleed traffic	CIC-IDS-2017 [54]	RF Classifier [54]	Out-of-distribution samples
Detect Malicious traffic (IDS)	CIC-IDS-2017 [54], Campus dataset	nPrintML [32]	Spurious correlations
Anomaly Detection	Mirai dataset [44]	Kitsune [44]	Out-of-distribution samples
OS Fingerprinting	CIC-IDS-2017 [54]	nPrintML [32]	Potential out-of-distribution samples
IoT Device Fingerprinting	UNSW-IoT [56]	Iisy [63]	Likely shortcut learning
Adaptive Bit-rate	HSDPA Norway [49]	Pensieve [42]	Potential out-of-distribution samples

Demonstrated prevalence of underspecification issues in existing ML artifacts for network security

Trustee in Action

IETF/IRTF Applied Networking Research Prize, 2023

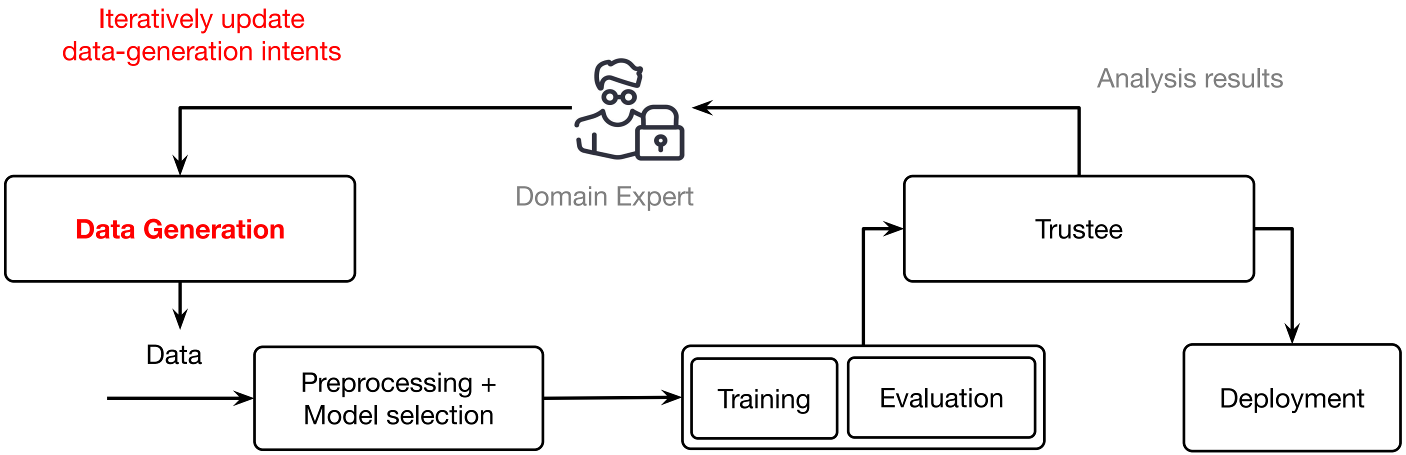
ACM CCS Best Paper Honorable Mention, 2022

Public Datasets

Problem	Dataset(s)	Model(s)	
Detect VPN traffic	Public VPN dataset [20]	1-D CNN [61]	Shortcut learning
Detect Heartbleed traffic	CIC-IDS-2017 [54]	RF Classifier [54]	Out-of-distribution samples
Detect Malicious traffic (IDS)	CIC-IDS-2017 [54], Campus dataset	nPrintML [32]	Spurious correlations
Anomaly Detection	Mirai dataset [44]	Kitsune [44]	Out-of-distribution samples
OS Fingerprinting	CIC-IDS-2017 [54]	nPrintML [32]	Potential out-of-distribution samples
IoT Device Fingerprinting	UNSW-IoT [56]	Iisy [63]	Likely shortcut learning
Adaptive Bit-rate	HSDPA Norway [49]	Pensieve [42]	Potential out-of-distribution samples

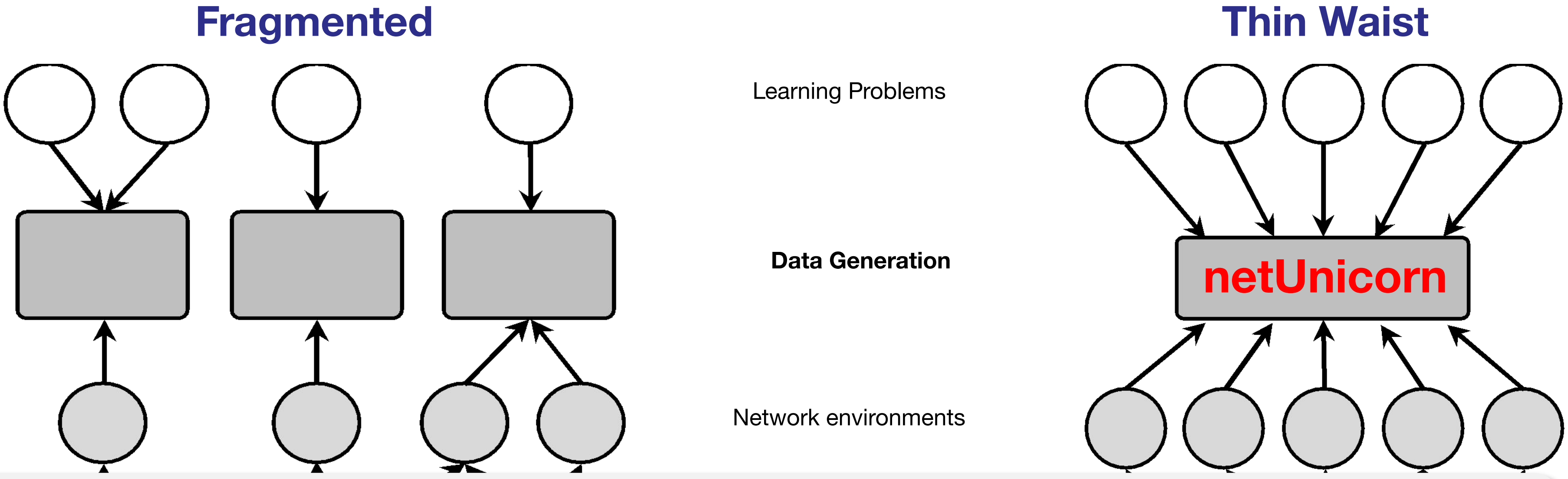
Attributed most underspecification issues to publicly available datasets

How to Fix the Underspecification (Data) Issues?



Use insights from model analysis to iteratively generate "better" data

How to Simplify Data Generation?



How to realize data-generation thin waist?

Disaggregation

- **Core Principle**

  A true thin waist emerges only when we decouple the layers that have historically been entangled

- **Disaggregate Intents from Execution**
  - Intent = what the experimenter wants: Execution = how it runs
  - Decoupling makes intents portable, reusable, and stable across environments

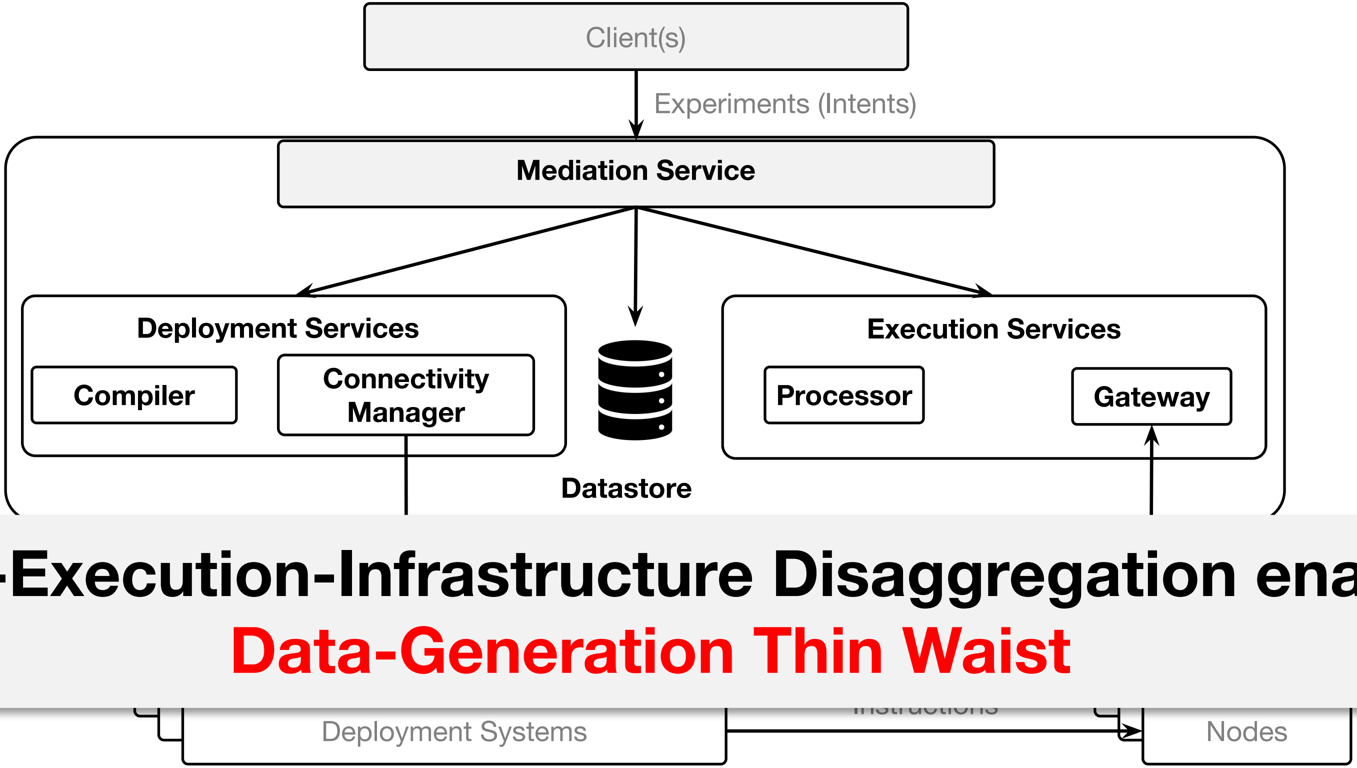
- **Disaggregate Execution from Infrastructure**

  Separation enables automatic mapping, capability-aware planning, and scaling across heterogeneity

Disaggregation In Action

Intents

Execution



Intent-Execution-Infrastructure Disaggregation enables
Data-Generation Thin Waist

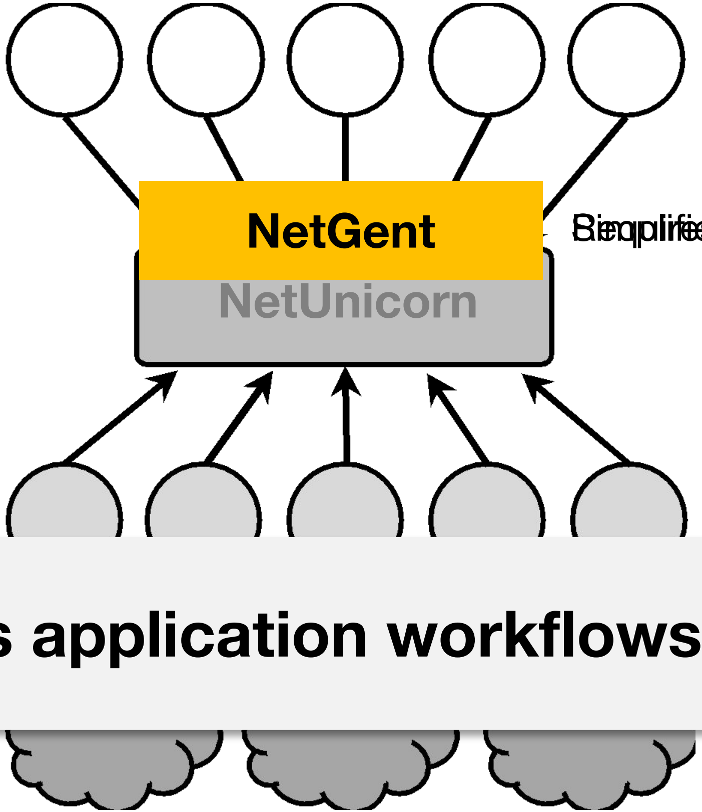
Infrastructure

Augmenting Data Generation “Thin Waist”

Learning Problems

Data Generation

Network environments



Simplifies synthesizing complex workflows
Requires manually coding app workflows

***NetGent*: Agent-Based Automation of Network Application Workflows**

Jaber Daneshamooz* Eugene Vuong‡ Laasya Koduru* Sanjay Chandrasekaran* Arpit Gupta*
*University of California Santa Barbara ‡California State University, East Bay

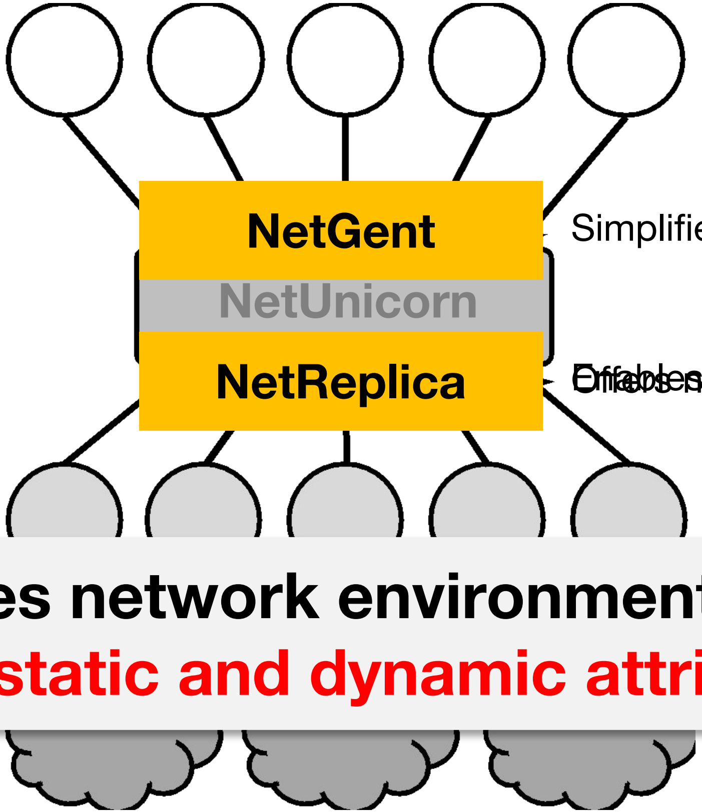
Disaggregates application workflows into reusable tasks

Augmenting Data Generation “Thin Waist”

Learning Problems

Data Generation

Network environments



NETREPLICA:
Toward a Programmable Substrate for Last-Mile Data Generation

Jaber Daneshamooz*, Satyandra Guthula*, Jessica Nguyen*, William Chen†, Sanjay Chandrasekaran*, Ankit Gupta‡, Arpit Gupta*, Walter Willinger§
University of California Santa Barbara* NIKSUN§ Cary Academy High School† AMD‡

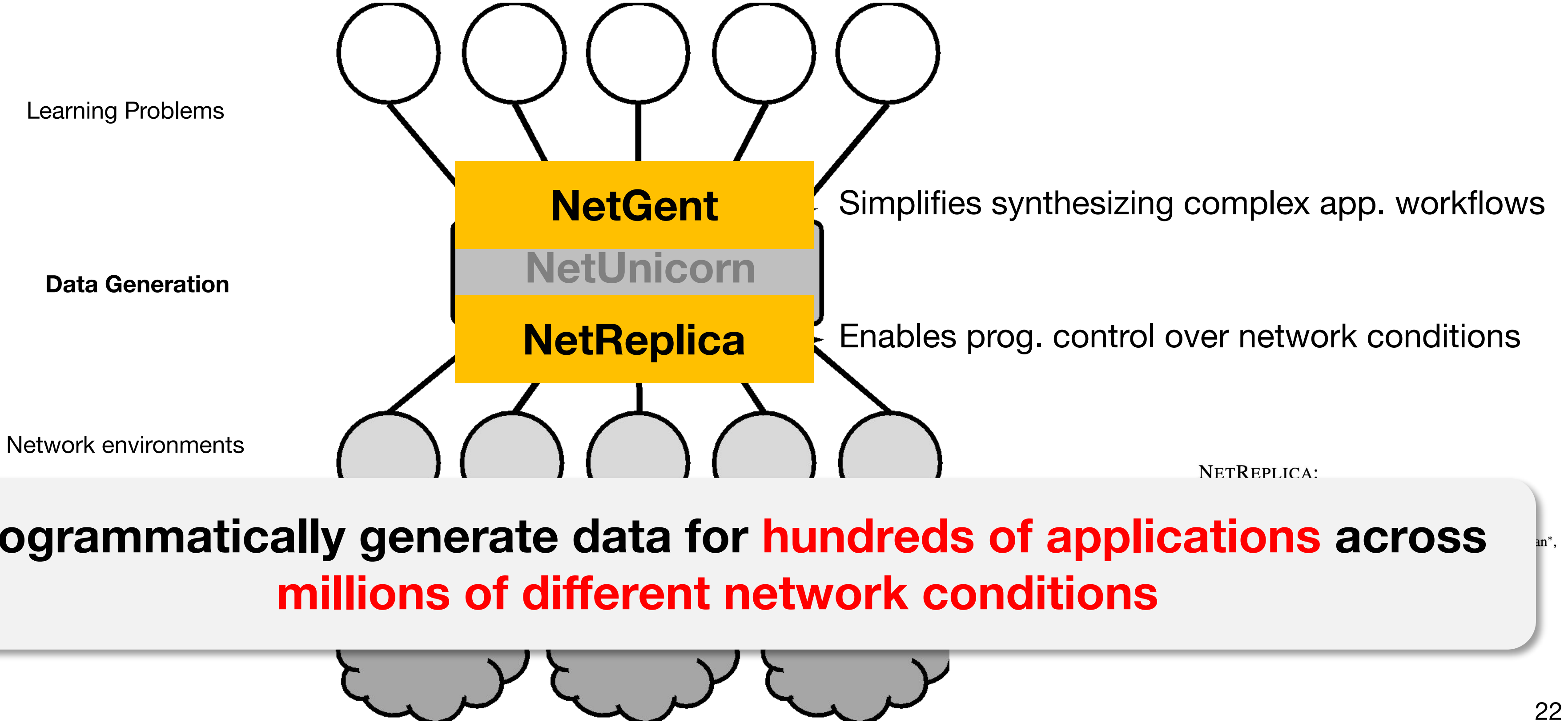
Simplifies synthesizing complex app. workflows

Offers no direct control over network conditions

Enables prog. control over network conditions

Disaggregates network environment specification into static and dynamic attributes

Augmenting Data Generation "Thin Waist"

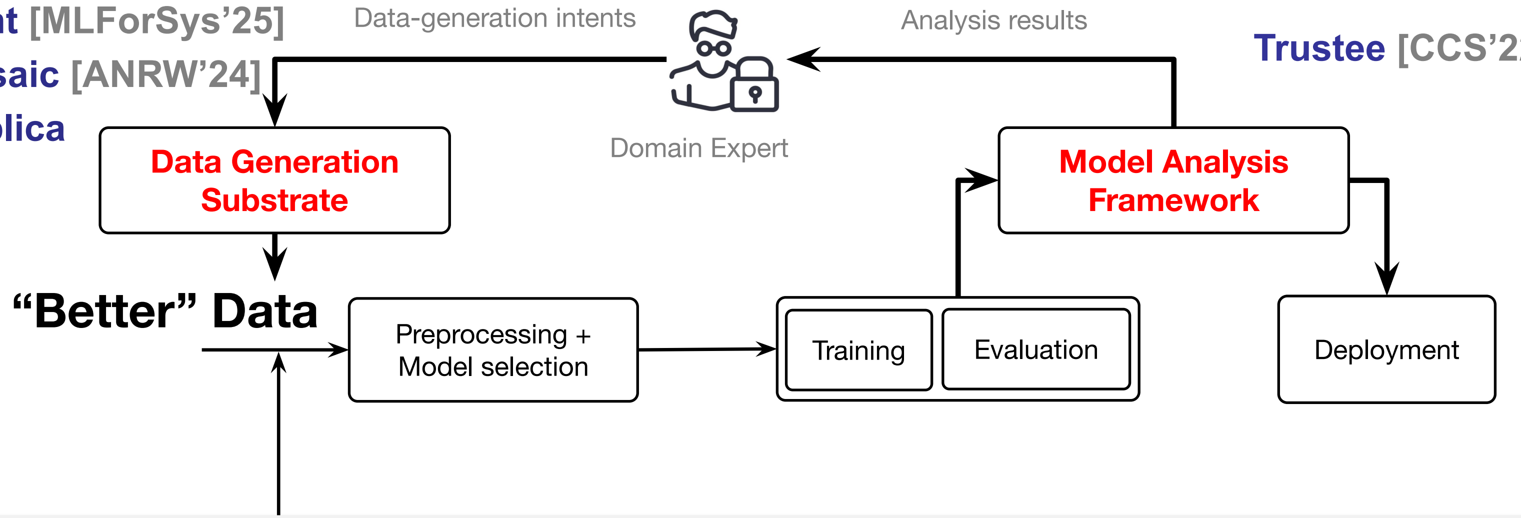


Programmatically generate data for hundreds of applications across millions of different network conditions

Closed-Loop ML Pipeline

NetUnicorn [CCS'23]
NetGent [MLForSys'25]
NetMosaic [ANRW'24]
NetReplica

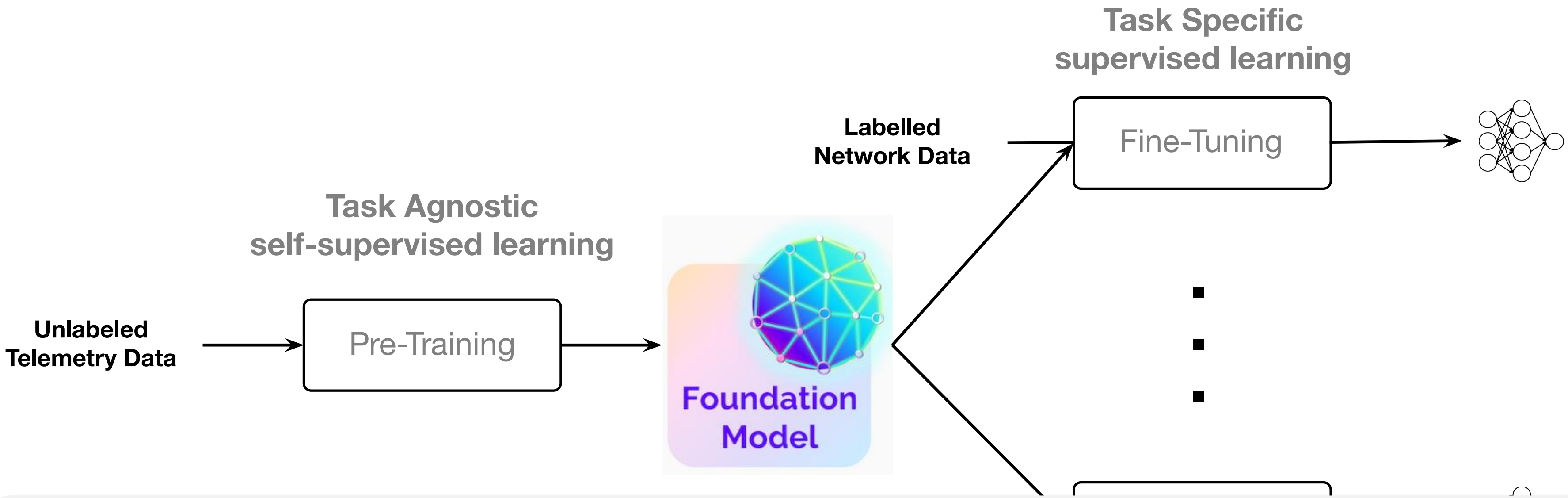
Trustee [CCS'22]



Only fixes generalizability for one model at a time

Data Generation Infrastructure

New Frontier: Network Foundation Models (NFMs)



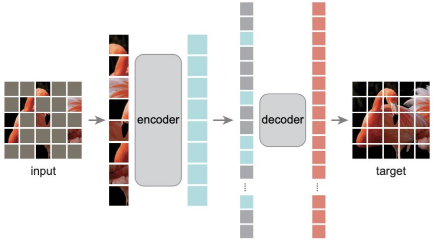
Critical to reason about generalizability of pre-trained Network Foundation Models

Thanks to SDN-powered telemetry infrastructure

Almost always the case, even with closed-loop ML pipeline

The Current Landscape for NFMs

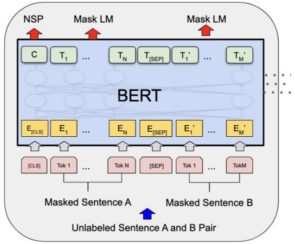
Masked Auto Encoders (MAE)



Image

YaTC

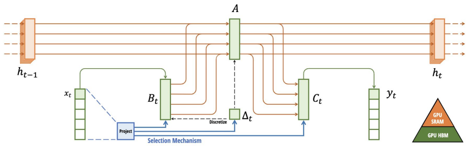
BERT



Natural Language

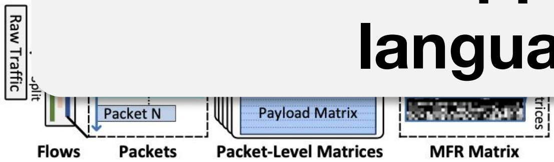
ET-BERT

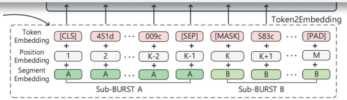
Mamba

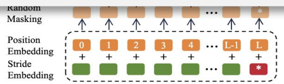


Linear State-Space

NetMamba







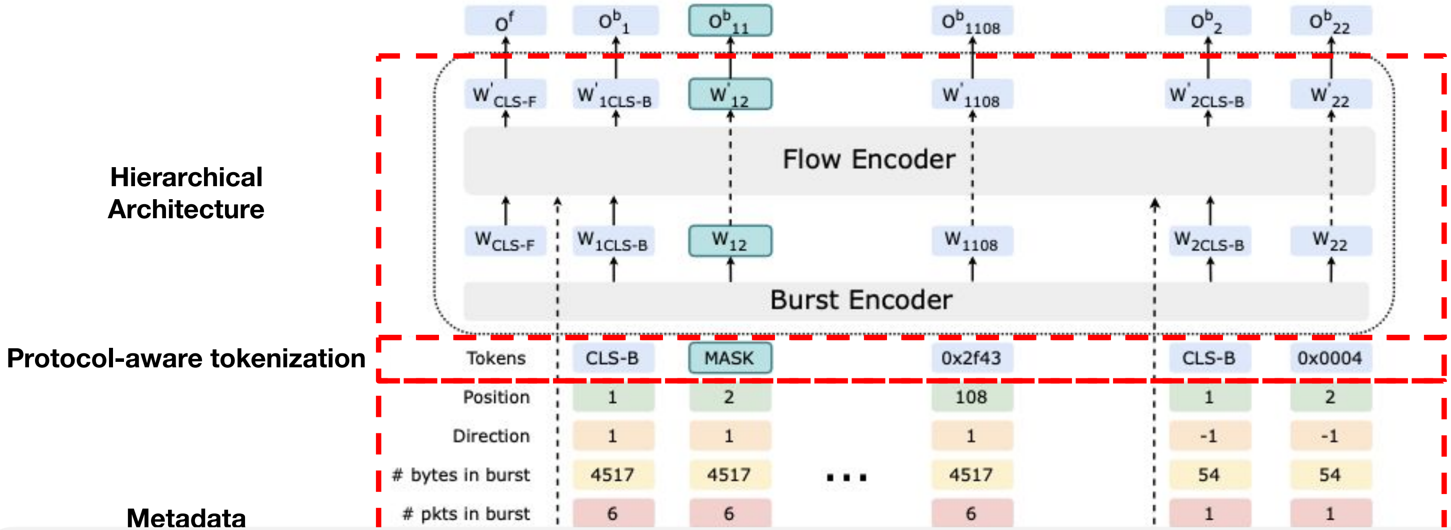
Force mapping of network data to images, natural language, or linear state-space models

What’s Unique about Network Data (Packet Traces)?

- **Embody different protocols (tokenization?)**
  Packet content is dictated by disparate protocols and standards
- **Entail variable length sequences (token selection?)**
  Packet sequences at any granularity is heavy tailed (multi-fractal behavior)
- **Encode multi-modal information (token embedding?)**
  Packet sequences carry critical spatial, temporal, & contextual information
- **Intrinsically hierarchical (modelling?)**
  Packets in different spatial/temporal groups have disparate semantic meanings

Necessitates a “Domain-specific” Approach

netFound: A Domain-Specific Network Foundation Model



How do we know if netFound is better, i.e., not only performant but also generalizable?

Standard Approach

			Deep Learning Models		Network Foundation Models		
Task	**Type**	**Dataset**	**Curtains (%)**	**nPrintML (%)**	**ET-BERT (%)**	**YaTC (%)**	**netFound (our) (%)**
1	Traffic Classification	Campus dataset	54.53 ± 0.97 $p < 0.001$	87.22 ± 0.12 $p < 0.001$	72.26 ± 0.38 $p < 0.001$	76.54 ± 0.23 $p < 0.001$	$\mathbf{96.08 \pm 0.04}$ –
2	Application Fingerprinting	Crossmarkets [63] (*Acc@10*)	20.64 ± 0.13 $p < 0.001$	64.83 ± 0.28 $p = 0.098$	35.62 ± 0.39 $p < 0.001$	58.13 ± 0.89 $p = 0.010$	$\mathbf{66.35 \pm 0.99}$ –
3	Application Fingerprinting	ISCXVPN-2016 [18]	66.85 ± 2.21 $p = 0.003$	84.10 ± 0.41 $p < 0.001$	77.57 ± 1.20 $p < 0.001$	83.84 ± 0.24 $p < 0.001$	$\mathbf{91.02 \pm 0.10}$ –
4	Intrusion Detection	CICIDS2017 [55]	99.75 ± 0.16 $p = 0.082$	99.93 ± 0.01 $p = 0.012$	99.94 ± 0.01 $p = 0.018$	99.92 ± 0.01 $p = 0.005$	99.99 ± 0.01 –
5	HTTP Bruteforce Detection	netUnicorn [11]	96.82 ± 0.22 $p = 0.006$	98.51 ± 0.02 $p < 0.001$	98.63 ± 0.02 $p < 0.001$	98.73 ± 0.10 $p = 0.030$	$\mathbf{99.01 \pm 0.01}$ –

Exhibits better performance for diverse "well-explored" learning tasks and datasets

Wait! Which Task? What Dat

The Sweet Danger of Sugar: Debunking Representation Learning for Encrypted Traffic Classification

Yuqi Zhao
Politecnico di Torino
Torino, Italy
yuqi.zhao@polito.it

Giovanni Dettori
Politecnico di Torino
Torino, Italy
giovanni.dettori@polito.it

Matteo Boffa
Politecnico di Torino
Torino, Italy
matteo.boffa@polito.it

Luca Vassio
Politecnico di Torino
Torino, Italy
luca.vassio@polito.it

Marco Mellia
Politecnico di Torino
Torino, Italy
marco.mellia@polito.it

	CIC-IDS (Heartbleed)		Crossmarket (*Acc@10*)	
ET-BERT	99.99 ± 0.01		99.82 ± 0.03	
YaTC	99.99 ± 0.01		99.69 ± 0.03	
netFound	99.99 ± 0.01		66.35 ± 0.99	
	Learning	Fixed	Learning	Fixed

Equating NFMs' success with performance on a limited set of downstream tasks and datasets is misleading

Our Approach: Intrinsic Evaluation Framework (IEF)

Assess embedding quality decoupled from downstream tasks/datasets

- **Embedding Geometry Analysis**
  How efficiently the model utilizes representation space?
- **Metric Alignment Assessment**
  How well it captures well-known features, developed by experts over time?
- **Causal Sensitivity Testing**
  How sensitive is the model to network context perturbations?

Experimental Setup

- **Network Foundation Models**

  YaTC, ET-BERT, NetFound, and NetMamba

- **Datasets**

  Endogenous (Actively Generated)

  Android Crossmarket, CIC-IDS, APT-IIoT

  Exogenous (Passive Traces)

  MAWI, CAIDA

Demystifying Network Foundation Models

Sylee (Roman) Beltiukov* UC Santa Barbara

Satyandra Guthula UC Santa Barbara

Wenbo Guo UC Santa Barbara

Walter Willinger NIKSUN, Inc

Arpit Gupta UC Santa Barbara

Embedding Geometry Analysis

- **Claim**

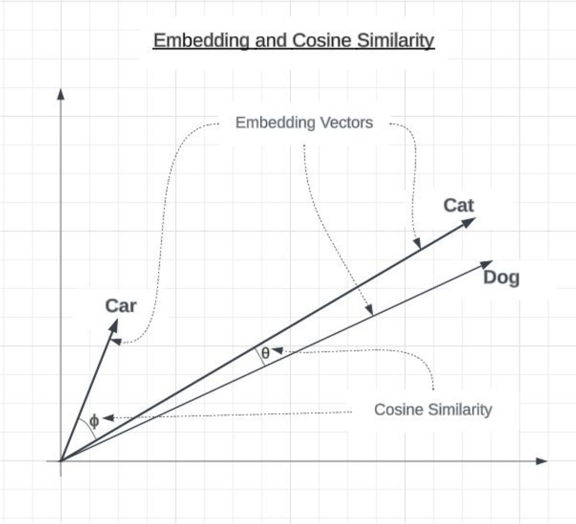
  Effective NFMs' embeddings should utilize the full representation space

- **Rationale**

  Well-distributed embedding geometry indicates that model distinguishes between flow along meaningful dimensions

- **Methodology**

  Measure anisotropy (cosine similarity) between embeddings pairs in the dataset



Example of embedding distribution in the space: *less related embeddings are further from each other*

Embedding Geometry Analysis Results

Dataset	YaTC	ET-BERT	netFound	NetMamba
	cos	*cos*	*cos*	*cos*
Crossmarket	0.85	0.88	0.69	0.93
CIC-APT-IIoT24	0.87	0.88	0.82	0.98
CIC-IDS2017	0.85	0.74	0.69	0.92
CAIDA	0.87	0.71	0.86	0.99
MAWI	0.88	0.78	0.94	0.99

We observe total representation collapse (NetMamba), dimensional dominance (YaTC), and limited generalizability (netFound)

Metric Alignment Assessment

- **Claim**

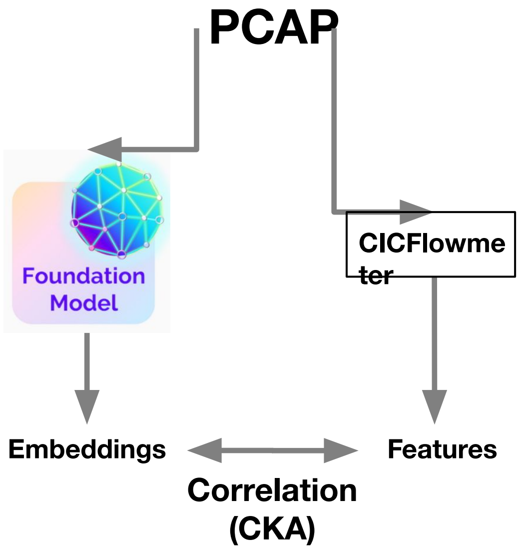
  Effective NFMs' should encode critical network statistics such as flow duration, packet size distributions, TCP dynamics, etc., without explicit supervision.

- **Rationale**

  These features, developed over decades, encode domain-knowledge, which worked well for different learning tasks

- **Methodology**

  Compute **CKA** similarity between each established metric and embeddings across flows



Metric Alignment Assessment

	Crossmarket	CIC-APT-IIoT24	CIC-IDS2017
YaTC	0.098	0.148	0.092
ET-BERT	0.012	0.014	0.064
netFound	**0.156**	**0.219**	**0.167**
NetMamba	0.047	0.141	0.042

Except netFound, most NFMs miss established metrics; yet netFound falters on production traces.

Result Summary: Intrinsic Evaluation Framework

- **Embedding Geometry Analysis**

  Total collapse (NetMamba), dimensional dominance (YaTC), limited generalizability (netFound)

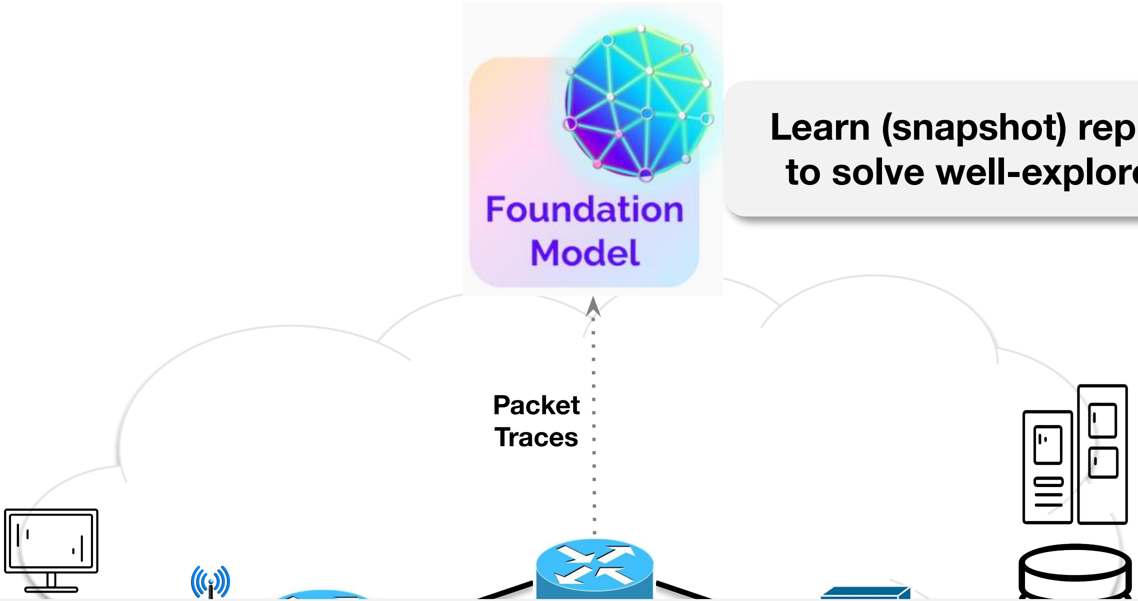
- **Metric Alignment Assessment**

  Poor alignment for most except netFound, which struggles to generalize

- **Causal Sensitivity Testing**

While no clear winner emerges, this framework provides an insightful benchmark

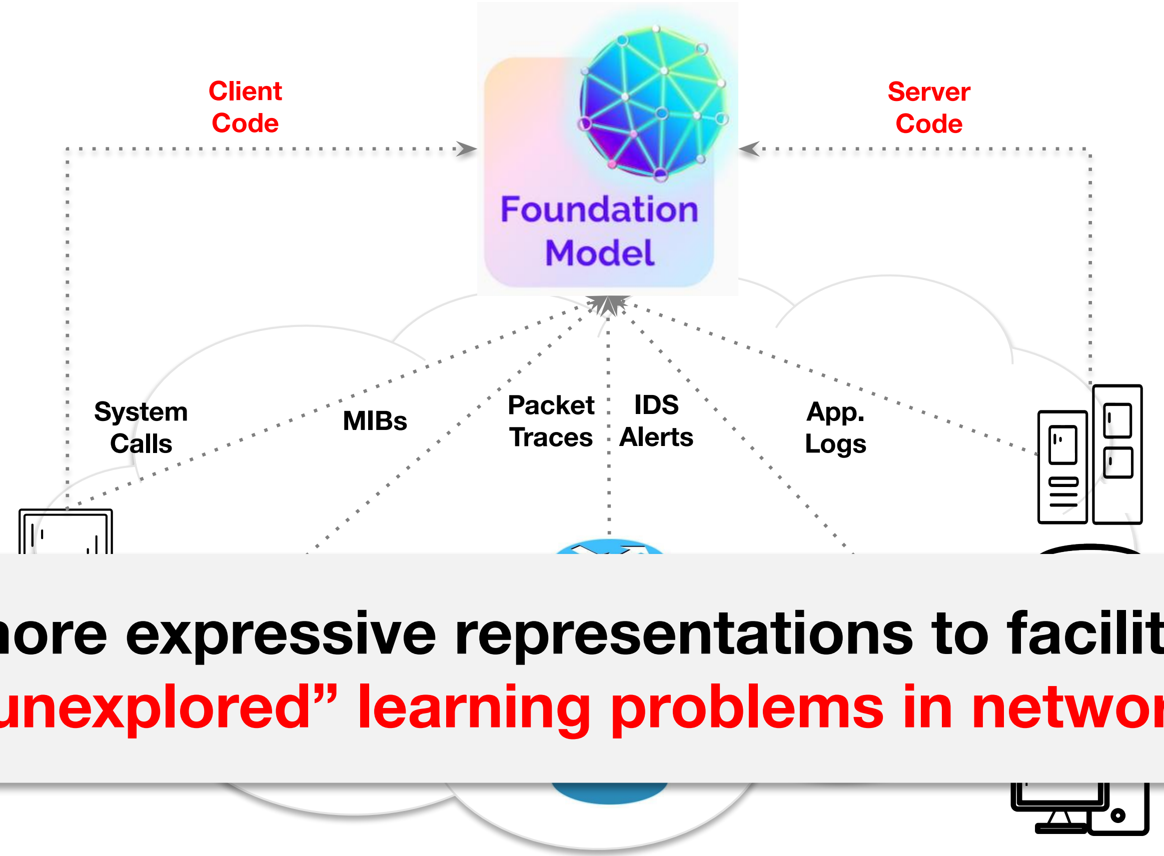
Current State



Learn (snapshot) representations for flows to solve well-explored learning problems

Can we learn more expressive and generalizable representations with least complexity and minimal data?

Where can we go from here?



Learn more expressive representations to facilitate solving "unexplored" learning problems in networking

"Unexplored" Learning Problems in Networking

- **Less Ambitious**

  Facilitate interactions with network data in natural language---rethink network telemetry systems

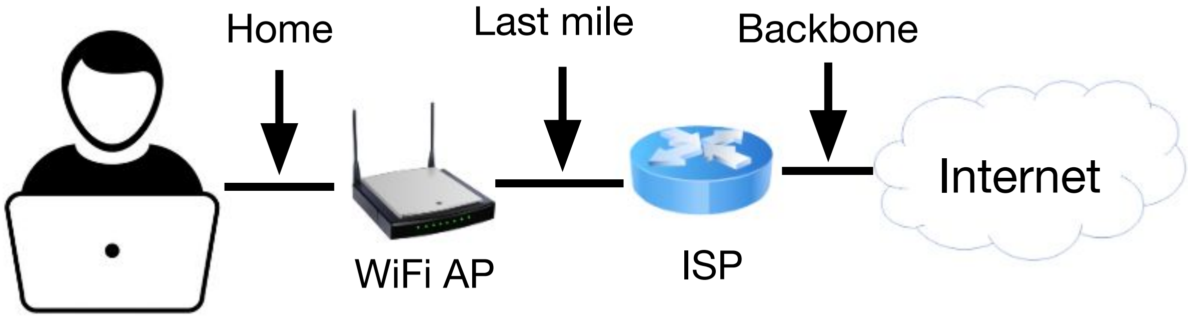
- **Moderately Ambitious**

  Identify and diagnose subtle and complex network events (anomalies), predict future events, and search/rank similar past events

Caution: Please remember the "garbage-in-garbage-out" adage and don't blindly throw LLMs for these problems

Code-2-pkt & pkt-2-code transformations for verification

Exemplary Use Case

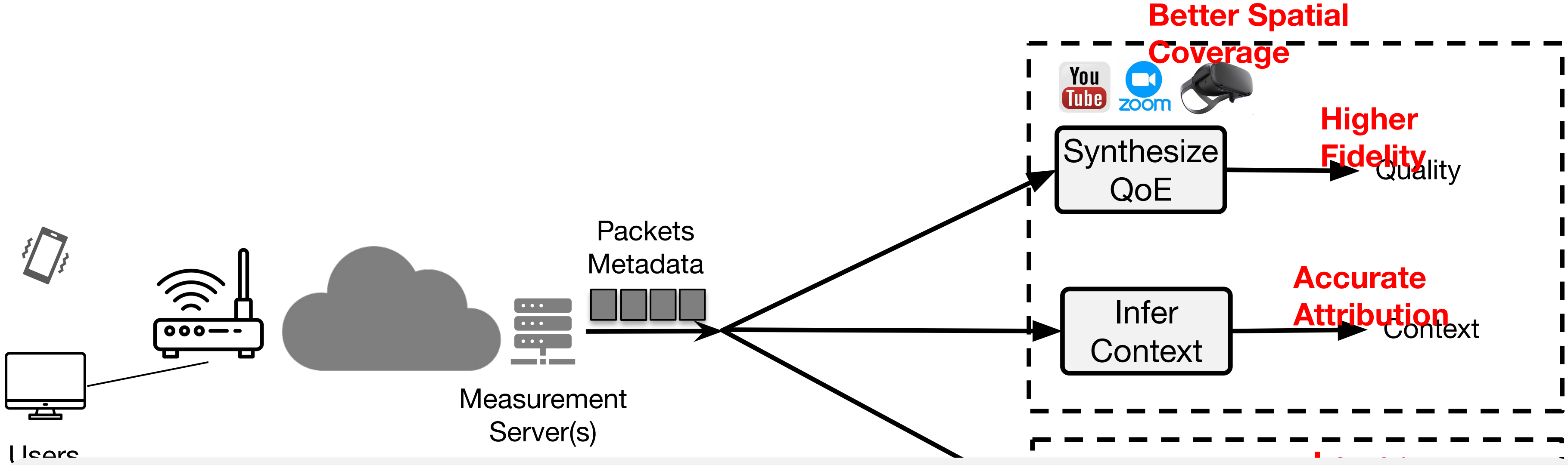


For every residence (i.e., **dense spatial coverage**), periodically (i.e., **dense temporal coverage**)

- Report metric(s) that captures users' quality of experience* (**high-fidelity assessment**)

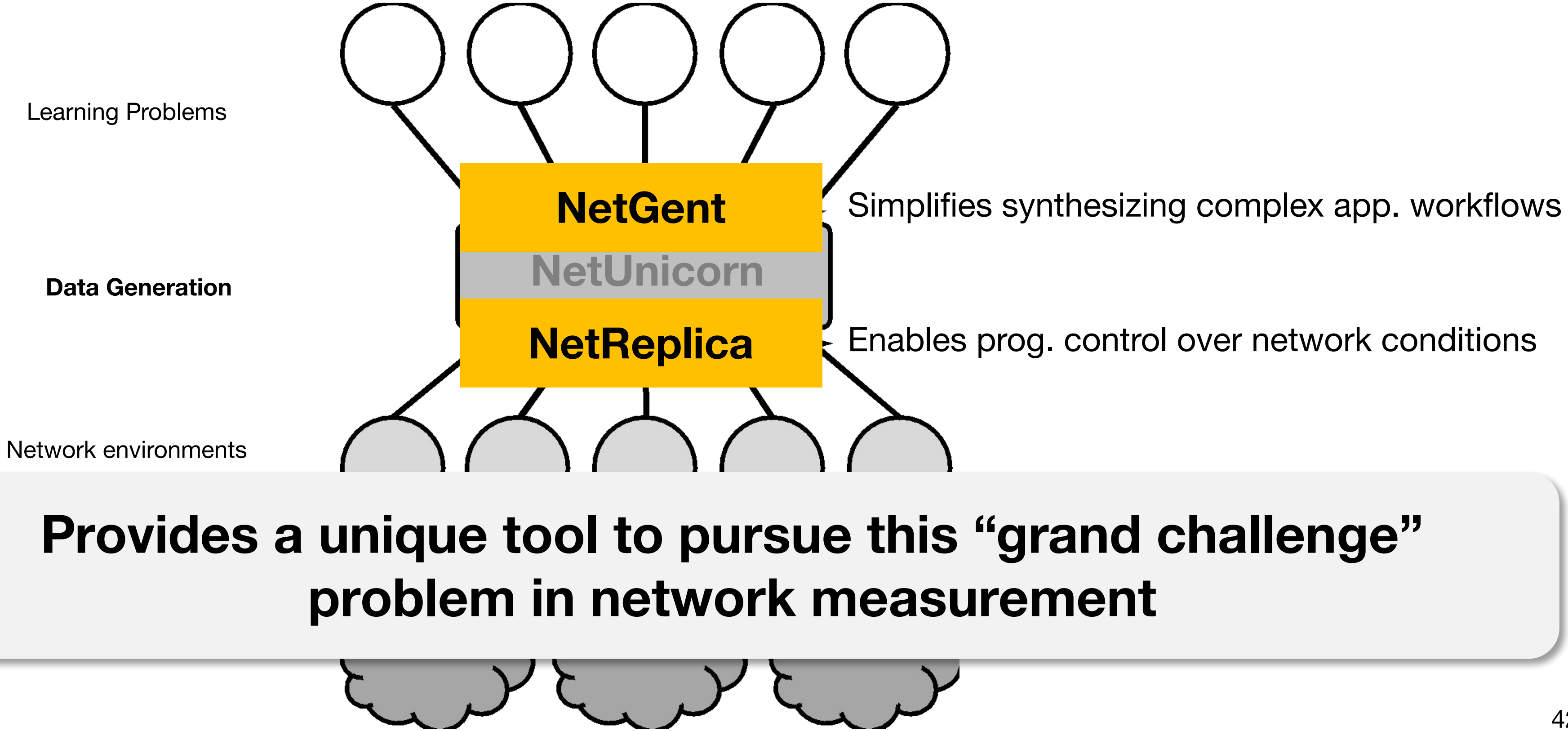
Existing tools fail to bridge gap between
measured and experienced broadband quality

Rethinking Broadband Quality Measurements



Learn data representations that enable closed-loop measurements, context inference, and counterfactual analysis

Programmable Data Generation Substrate



Provides a unique tool to pursue this “grand challenge” problem in network measurement

Summary

NeurIPS' 25

Demystifying Network Foundation Models

Sylee (Roman) Beltiukov*
UC Santa Barbara

Satyandra Guthula
UC Santa Barbara

Wenbo Guo
UC Santa Barbara

Walter Willinger
NIKSUN, Inc

Arpit Gupta
UC Santa Barbara

- Production-ready ML models critical for **self-driving networks**
- **Closed-loop ML pipeline,** composed of model-analysis (**Trustee**) and data-generation (**NetUnicorn**, **NetReplica**, **NetGent**) substrates critical for production-ready ML
- **Network foundation model(s)** are critical for self-driving networks
  - **Intrinsic evaluation framework** helps answer what a Network foundation model is (and is not) learning
  - **Programmable data-generation substrate** provide a unique opportunity to take on new and revisit older problems with a fresher perspective