Defeating Slow-and-Low Threats via Diffusion Model-Based Generative Inference

Seyed Mohammad Mehdi Mirnajafizadeh*

Prashant Khanduri*, DaeHun Nyang†, Rhongho Jang*

* WAYNE STATE UNIVERSITY

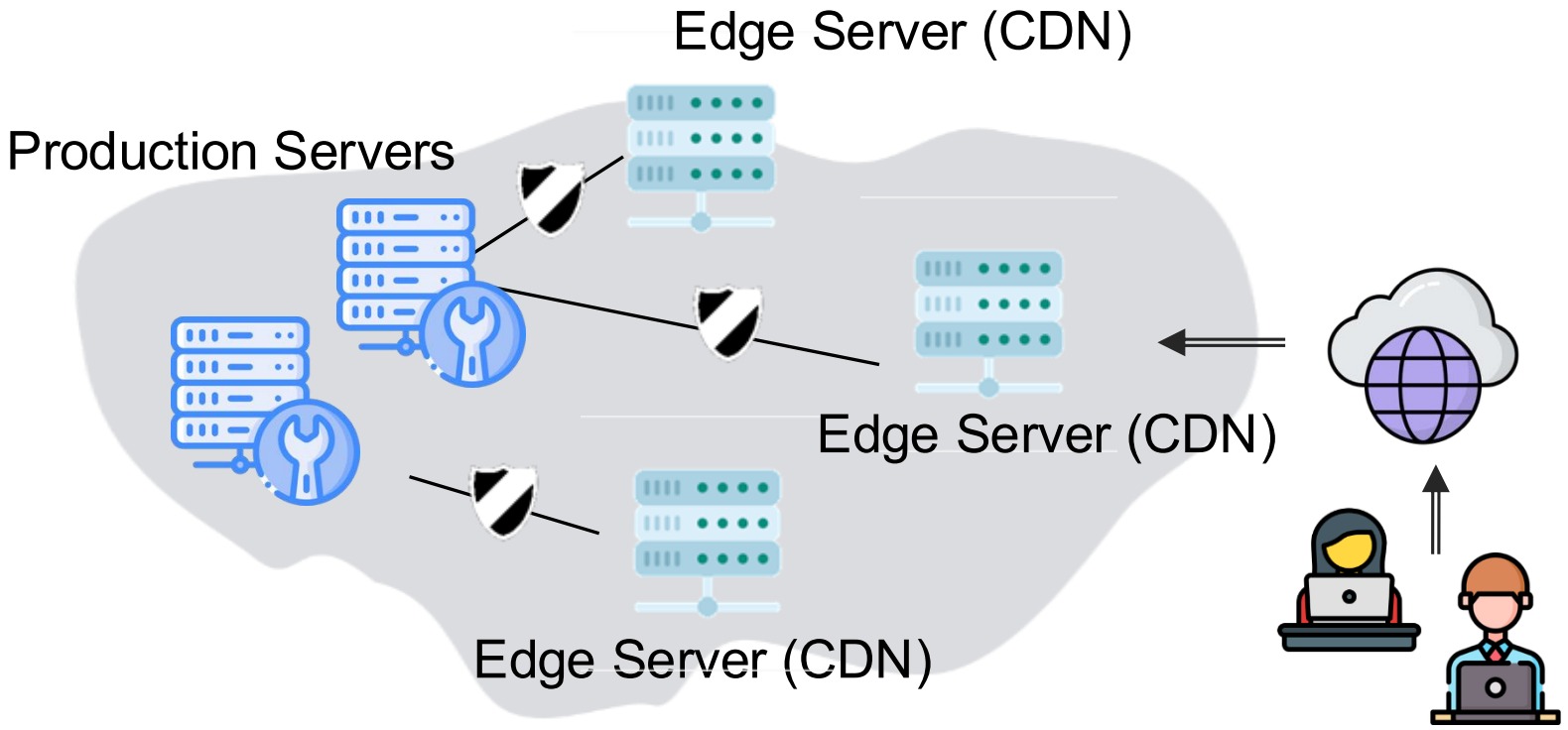
† 이화여자대학교 EWHA WOMANS UNIVERSITY

USENIX NSDI Symposium 2026

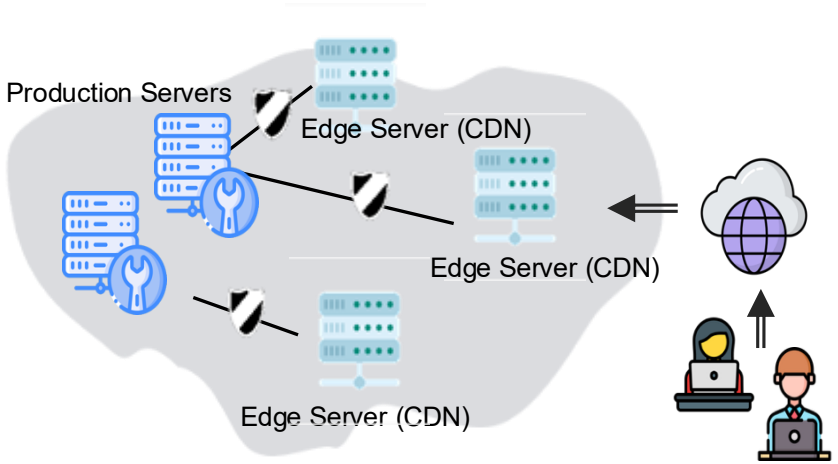
Outline

- Background
- Motivation
- Proposed System
- Evaluation
- Conclusion

Background: CDN Defense Constraints



Background: CDN Defense Constraints

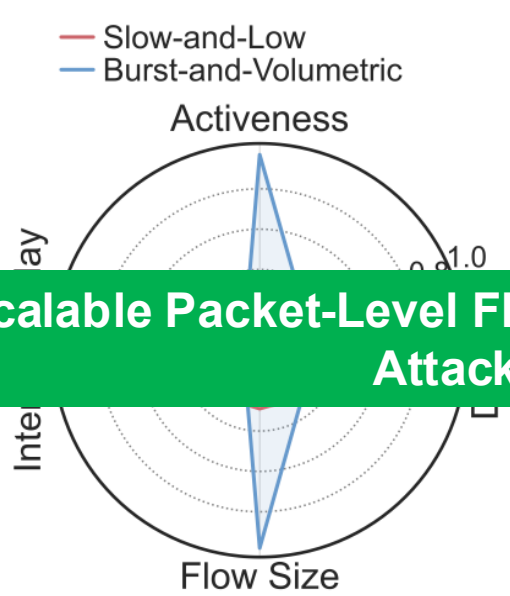


- Primary Defense Layer
  - First-line shield for production servers
- Performance-First Design
  - Optimized for high throughput and low latency

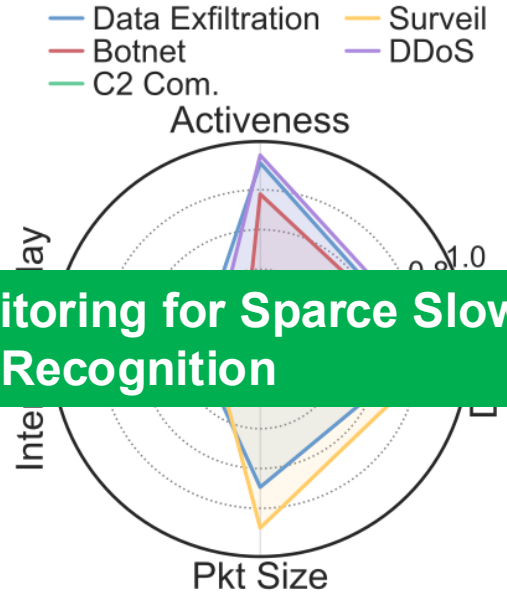
Short Monitoring Windows → **Slow-and-Low Evasion**

Motivation: Slow-and-Low Attack Analysis

- **80+ attack type analysis**
  - **Burst-and-Volumetric:** high volume, high burstiness and activity over short duration
  - **Slow-and-Low:** lower volume and activity over much longer duration



Attack Categories



Slow-and-Low Attacks

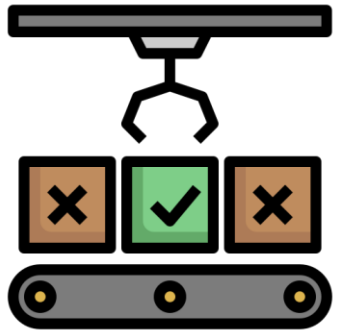
A Scalable Packet-Level Flow Behavior Monitoring for Sparce Slow-and-Low Attack with Multi-Class Recognition

Motivation: Challenges

C1: Lacking Scalable Flow Behavior Monitor

- Unbounded memory overhead risks defense-level resource saturation

Unbounded Per-Flow
Packet-Level Monitoring



Bounded Selective
Flow Monitoring

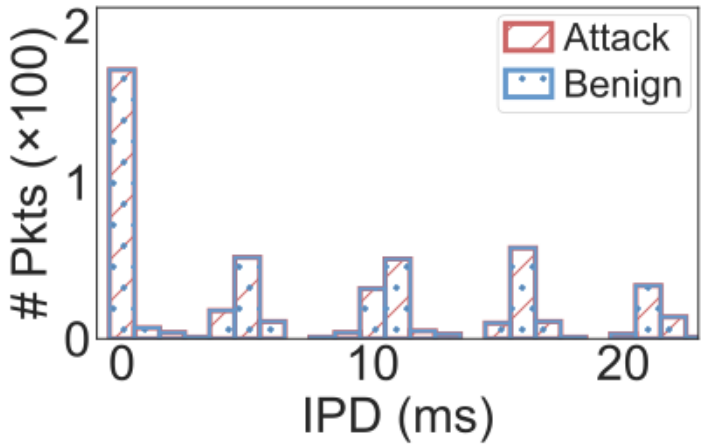
Motivation: Challenges

C1: Lacking Scalable Flow Behavior Monitor

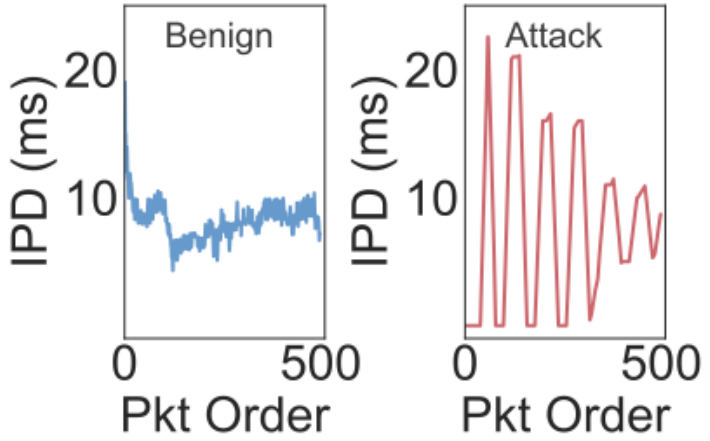
- Unbounded memory overhead risks defense-level resource saturation

C2: Evading Coarse Behavior-Based Detection

- Feature aggregation saves resources but enables attack mimicry



Flow-Level Distribution



Packet-Level Temporal Behavior

Motivation: Challenges

C1: Lacking Scalable Flow Behavior Monitor

- Unbounded memory overhead risks defense-level resource saturation

C2: Evading Coarse Behavior-Based Detection

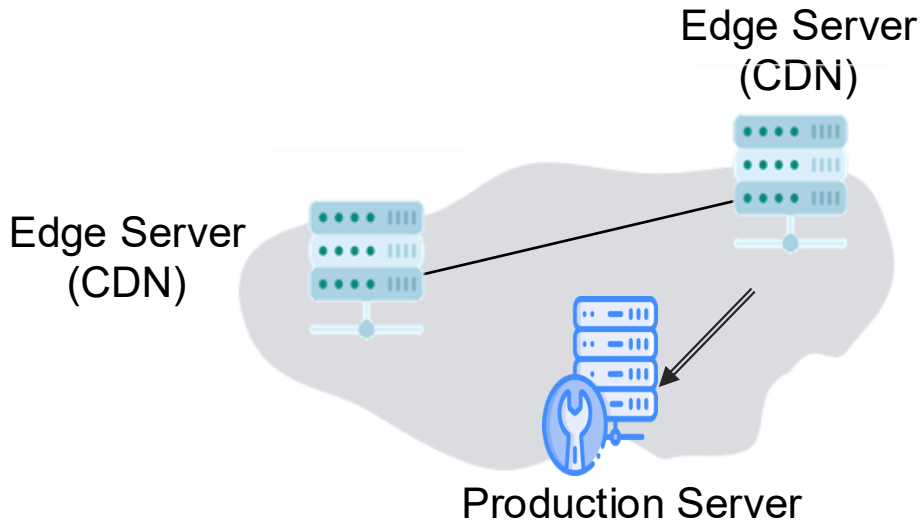
- Feature aggregation saves resources but enables attack mimicry

C3: Partial View of Slow-and-Low Attacks

- State-of-the-art rely on hard timeout or threshold-based heuristics
- Flow features exhibit different behaviors at different stages

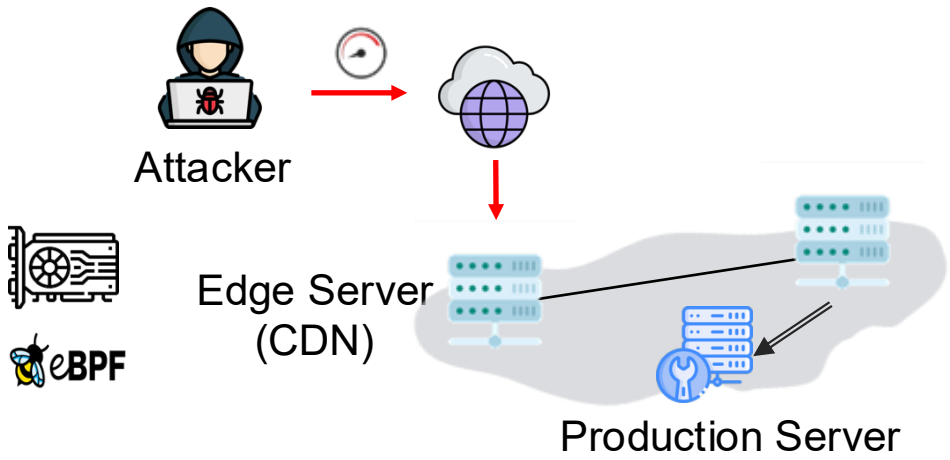
SketchVision

- Designed for deployment at CDN edge servers to protect against slow-and-low attacks
- SketchVision operates
  - Via an eBPF monitoring module in kernel space
  - With a consumer-grade GPU for AI-assisted tasks

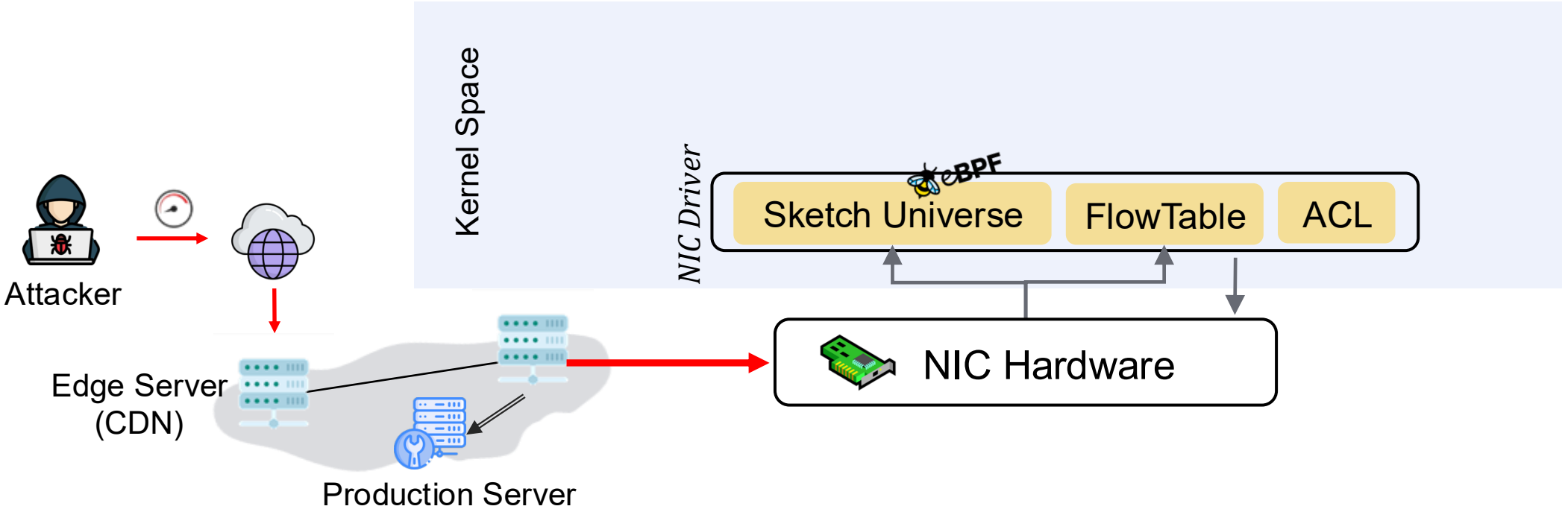


SketchVision

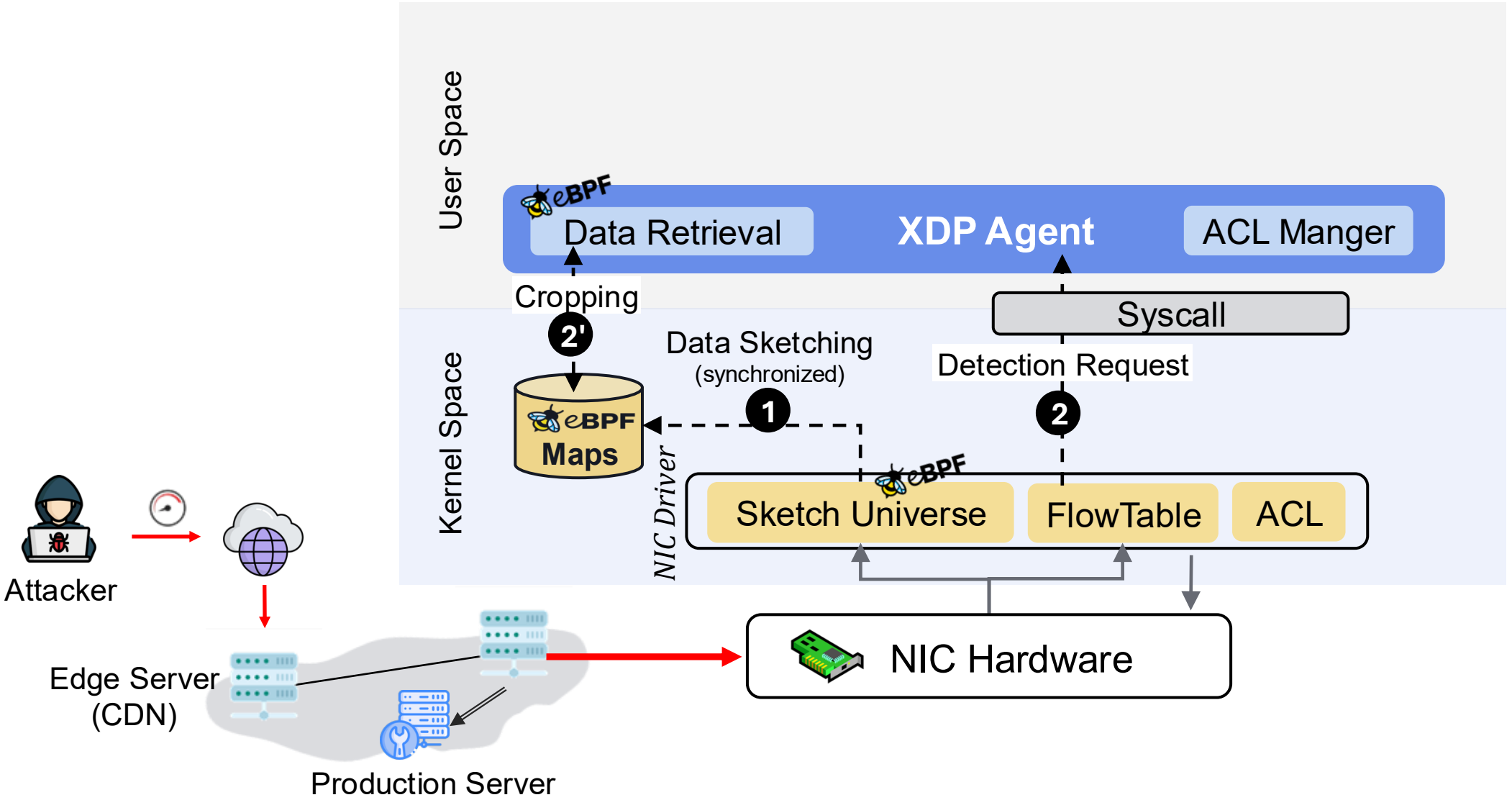
- Designed for deployment at CDN edge servers to protect against slow-and-low attacks
- SketchVision operates
  - eBPF monitoring module in kernel space
  - An edge GPU for AI-assisted tasks



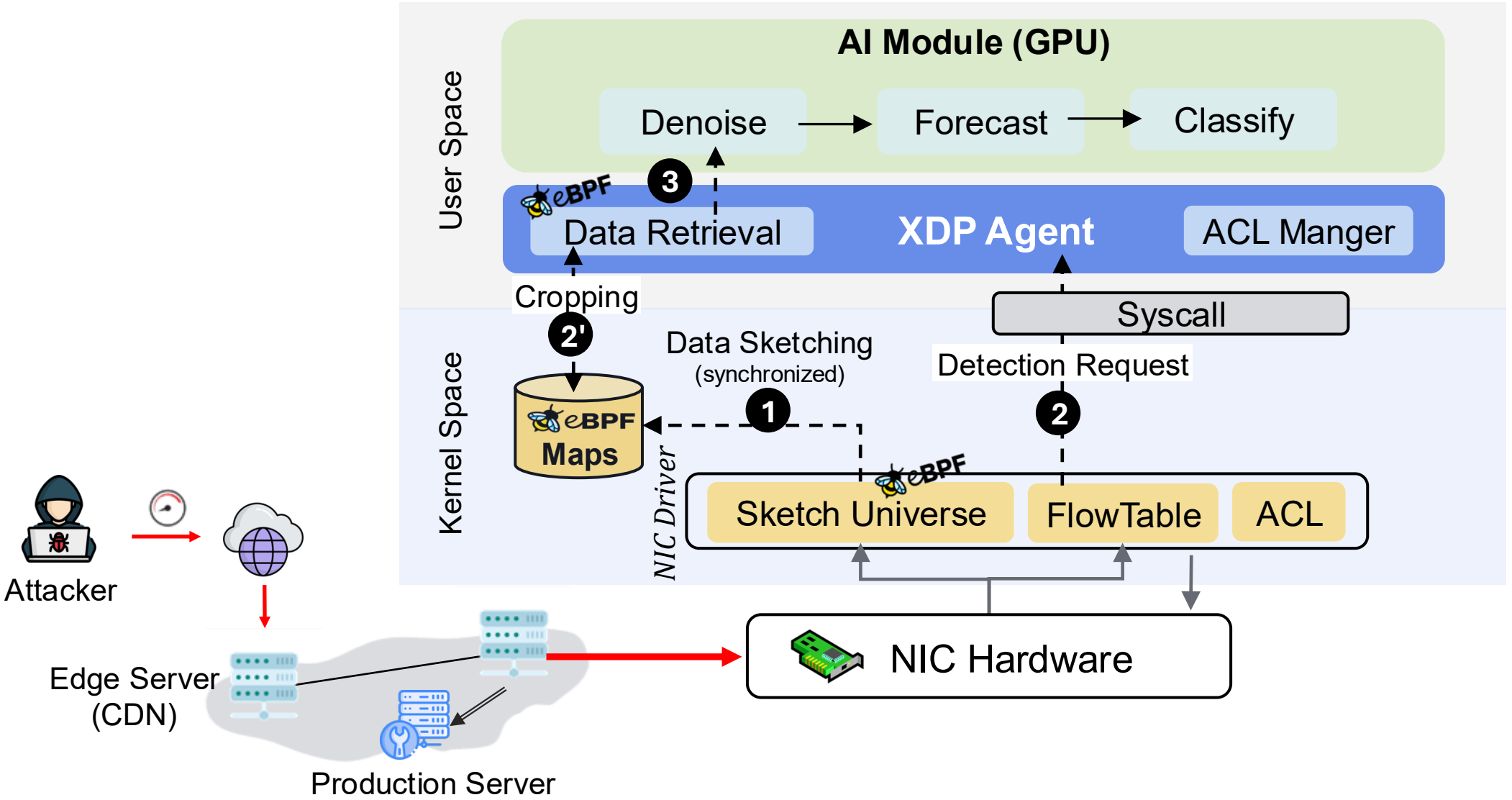
SketchVision



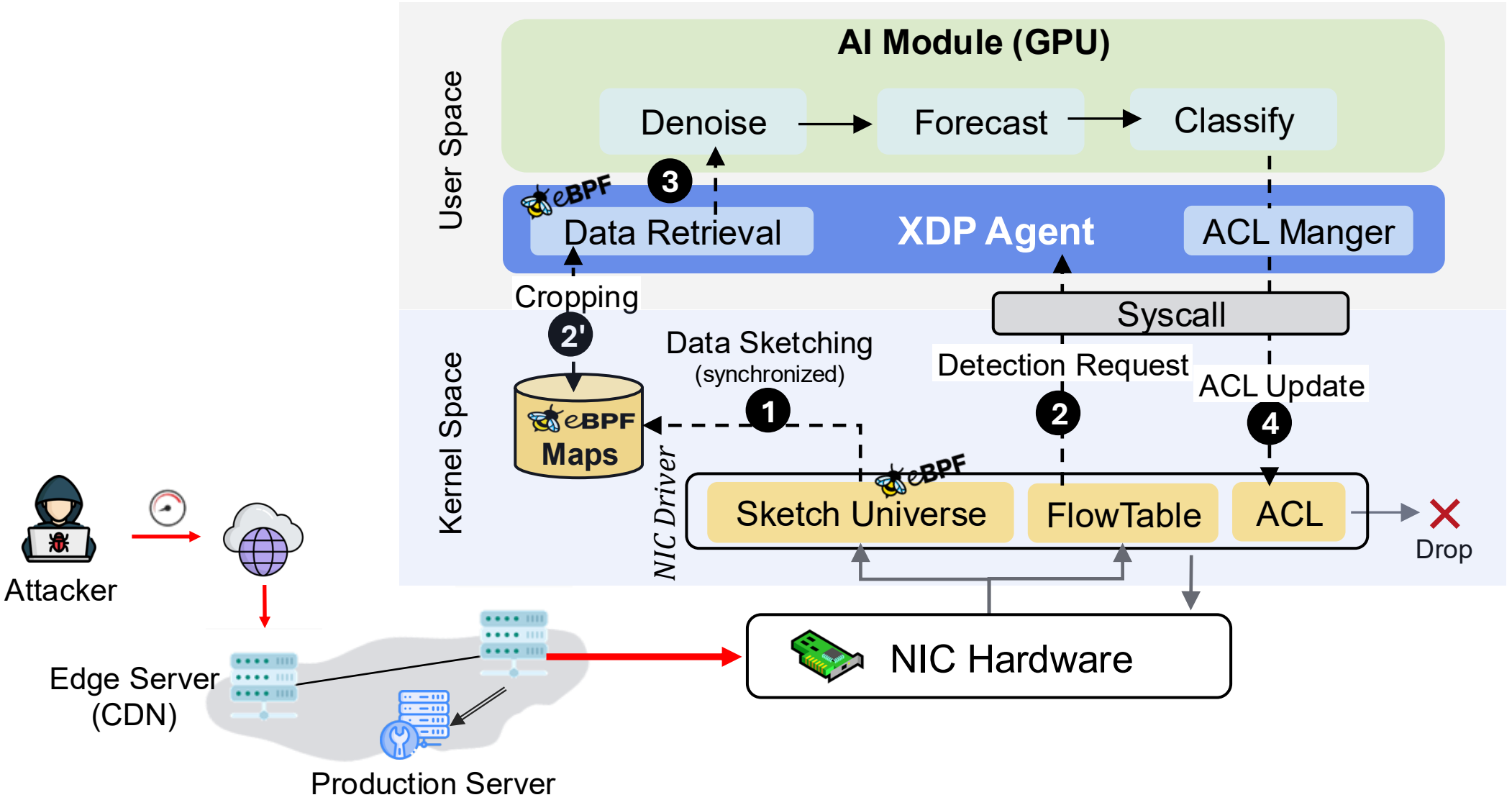
SketchVision



SketchVision

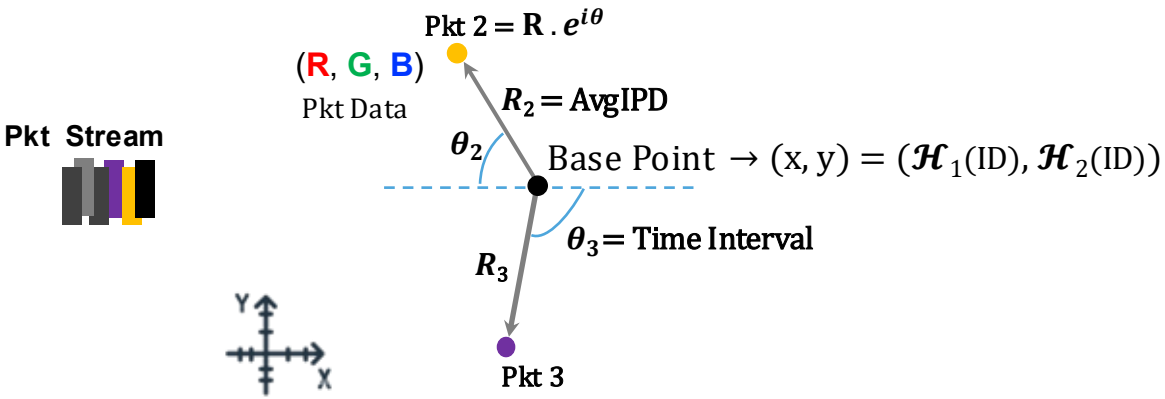


SketchVision



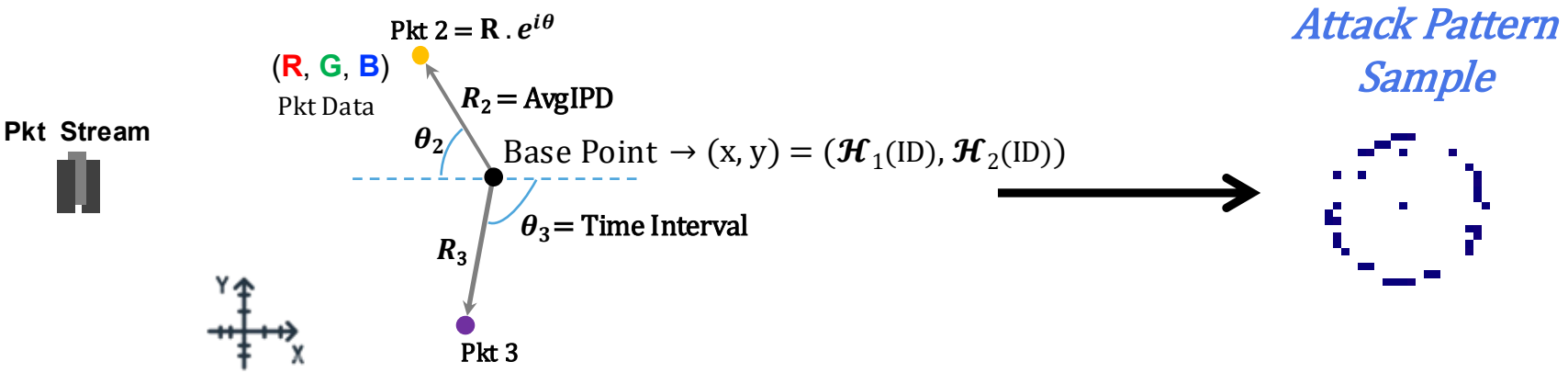
Vision-Inspired Behavior Sketching: Flow Painting

- SketchVision extends
  - Per-flow counting to vision-based flow representation
- Base point derived via independent hash pairs
  - Packets' location derived via polar coordinates (θ, R)
  - Metadata embedded into the location

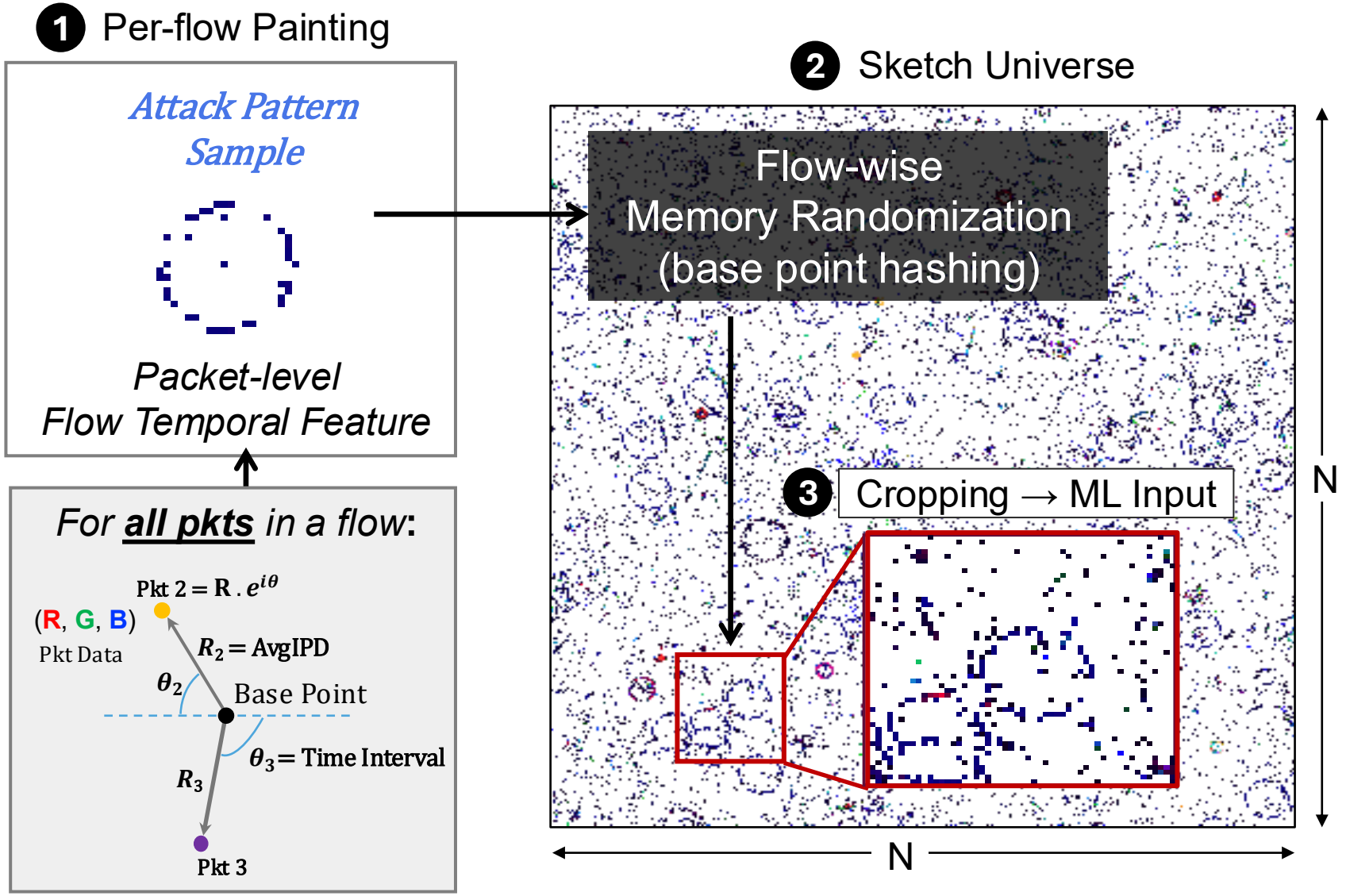


Vision-Inspired Behavior Sketching: Flow Painting

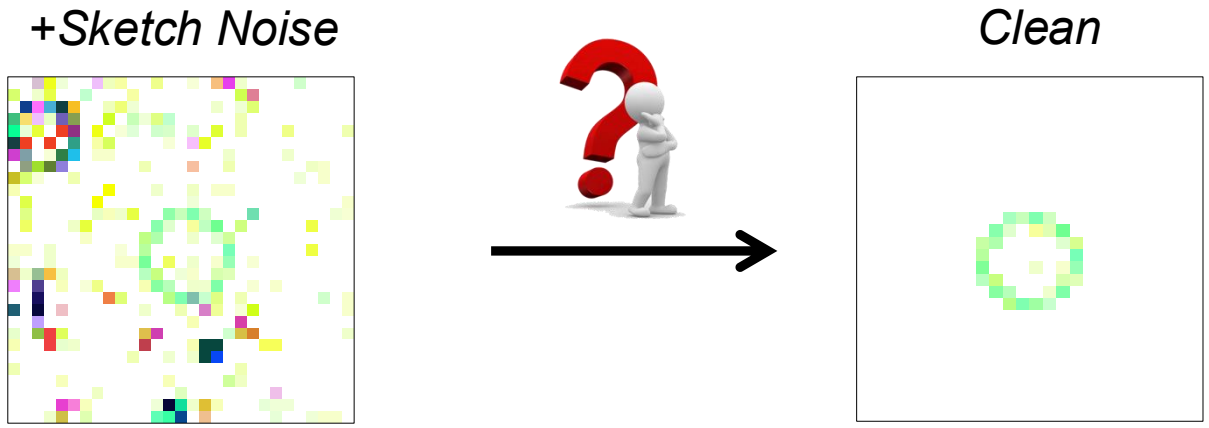
- SketchVision extends
  - Per-flow counting to vision-based flow representation
- Progressively painting a visual flow's behavior representation over time



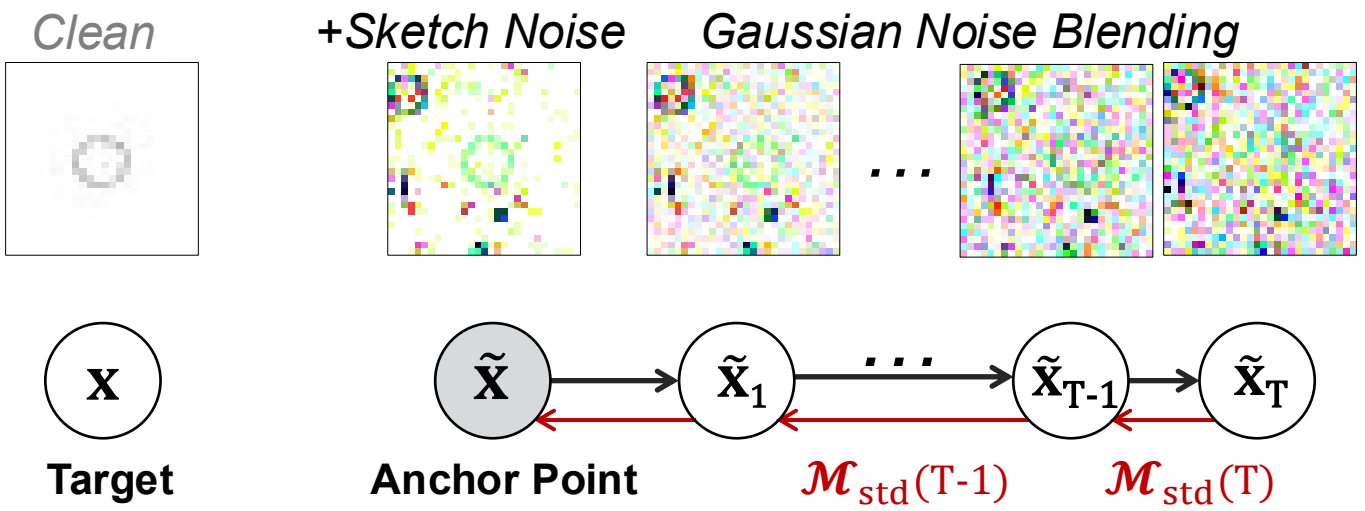
Vision-Inspired Behavior Sketching



Diffusion Model as Denoiser

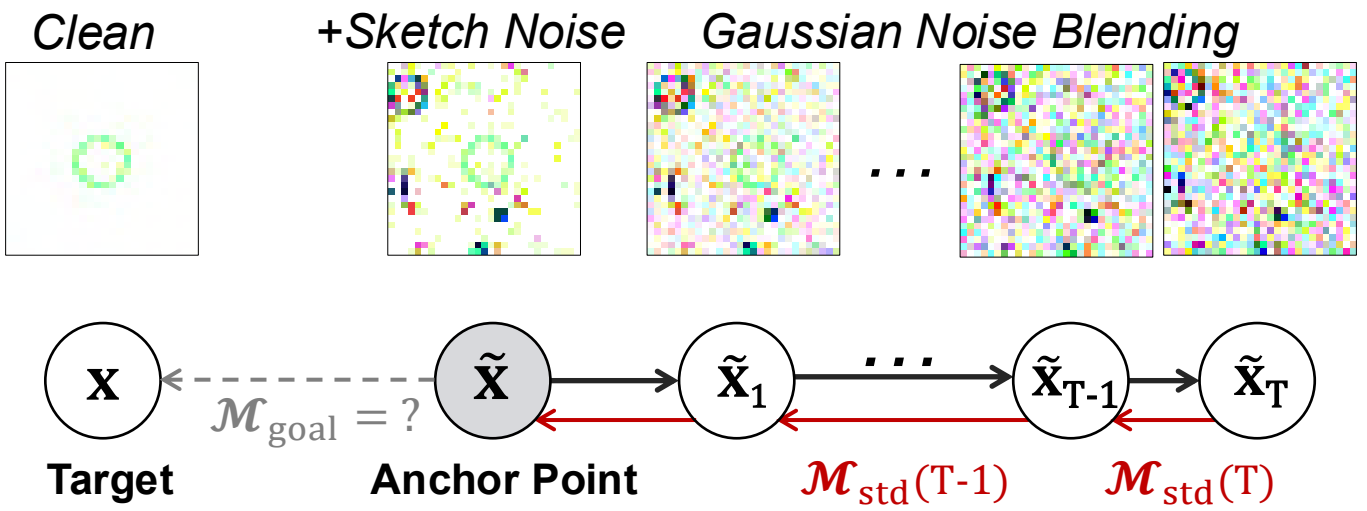


Diffusion Model as Denoiser

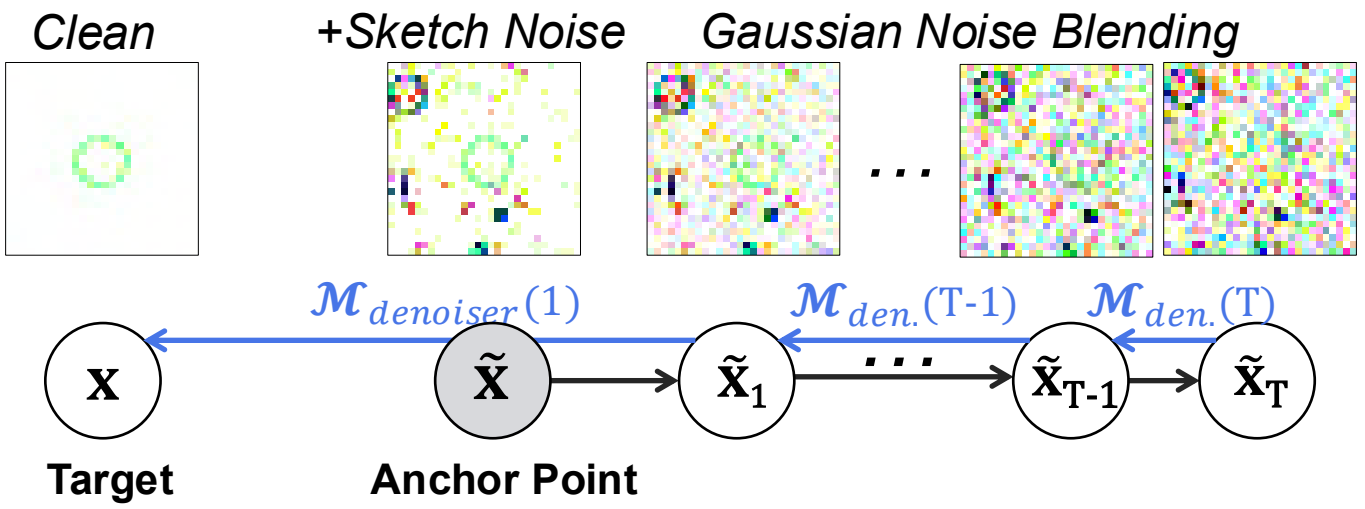


$$\mathcal{L} = \mathbb{E}_{x,t,\varepsilon,c}\left[\left\|\varepsilon_\theta\left(\tilde{\mathrm{X}}_t, t, c\right) - \tilde{\mathrm{X}}_t\right\|^2\right]$$

Diffusion Model as Denoiser



Diffusion Model as Denoiser

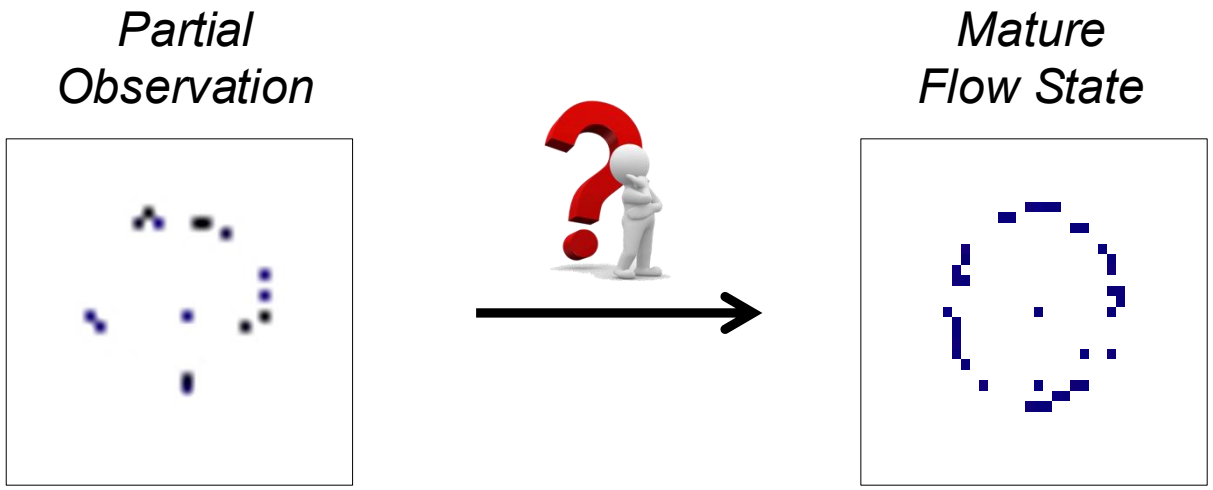


$\tilde{\mathbf{X}}^p \rightarrow \mathbf{X}^p$

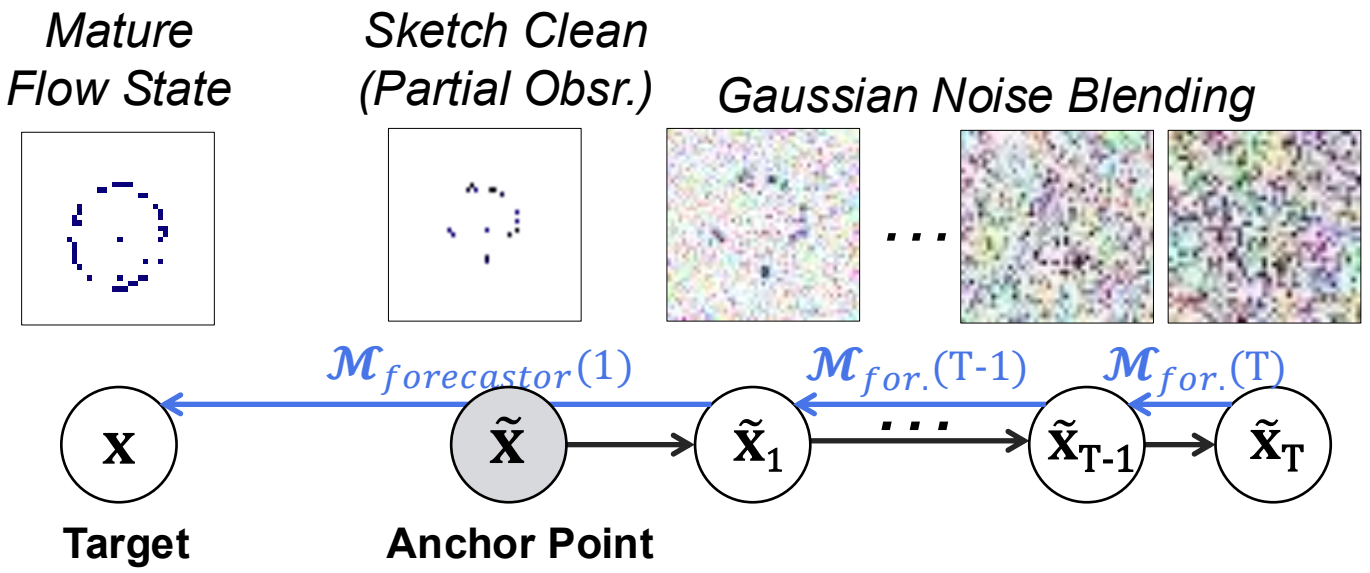
$$\mathcal{L} = \mathbb{E}_{x,t,\varepsilon,c}\left[\left\|\varepsilon_\theta\left(\tilde{\mathrm{X}}_t, t, c\right) - \mathrm{X}\right\|^2\right]$$

$c = (label, \#pkt)$

Diffusion Model as Forecastor



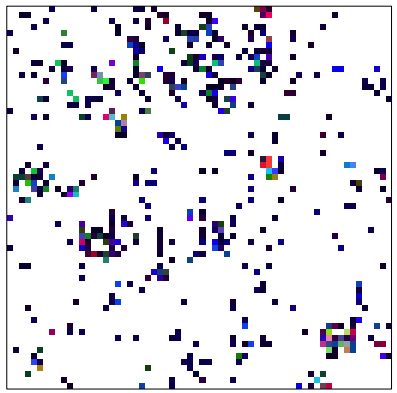
Diffusion Model as Forecastor



$$\mathbf{X}^p \rightarrow \mathbf{X}$$

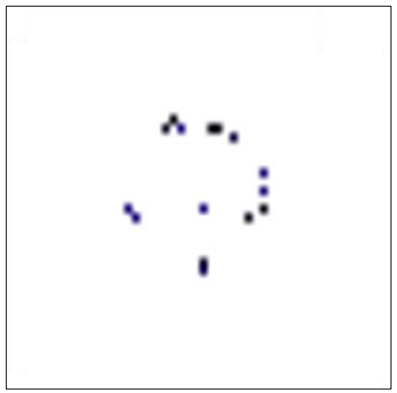
$$\mathcal{L} = \mathbb{E}_{x,t,\varepsilon,c}\left[\left\|\varepsilon_\theta\left(\mathrm{X}_t^p, t, c\right) - \mathrm{X}\right\|^2\right]$$

Diffusion Model as Denoiser & Forecastor

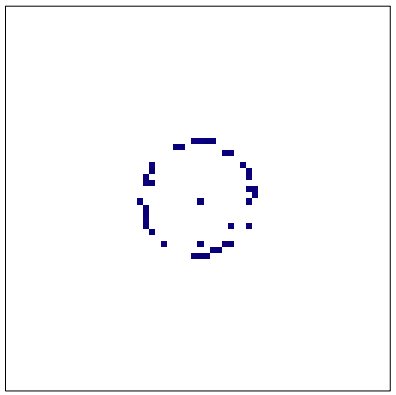


Sketch Noise

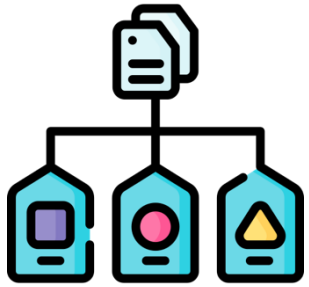
Denoise



Forecast



Classify



$attack_1$ ⋯ $benign$

Experimental Highlight

- Implementations:
  - Kernel Space: eBPF XDP kernel for flow sketching
  - User Space: XDP agente bridging in-kernel sketching and ML
  - AI Model: Diffusion model + multi-classfier CNN
- Testbed:
  - CPU: Inetl® Xeon® 8358 (64 cores / 128 threads, 2.60 GHz)
  - Memory: 1TB DRAM
  - GPU: NVIDIA A100
- Dataset and ML training:
  - 19 types of slow attack in 5 categories
    - Botnet, C2 Communication, DDoS Data Exfiltration, Surveillance
    - Trained across flow progression using (clean, noisy) images + corresponding mature flow state

Early Detection via Generative Inference

(Denoise → Forecast → Classify)

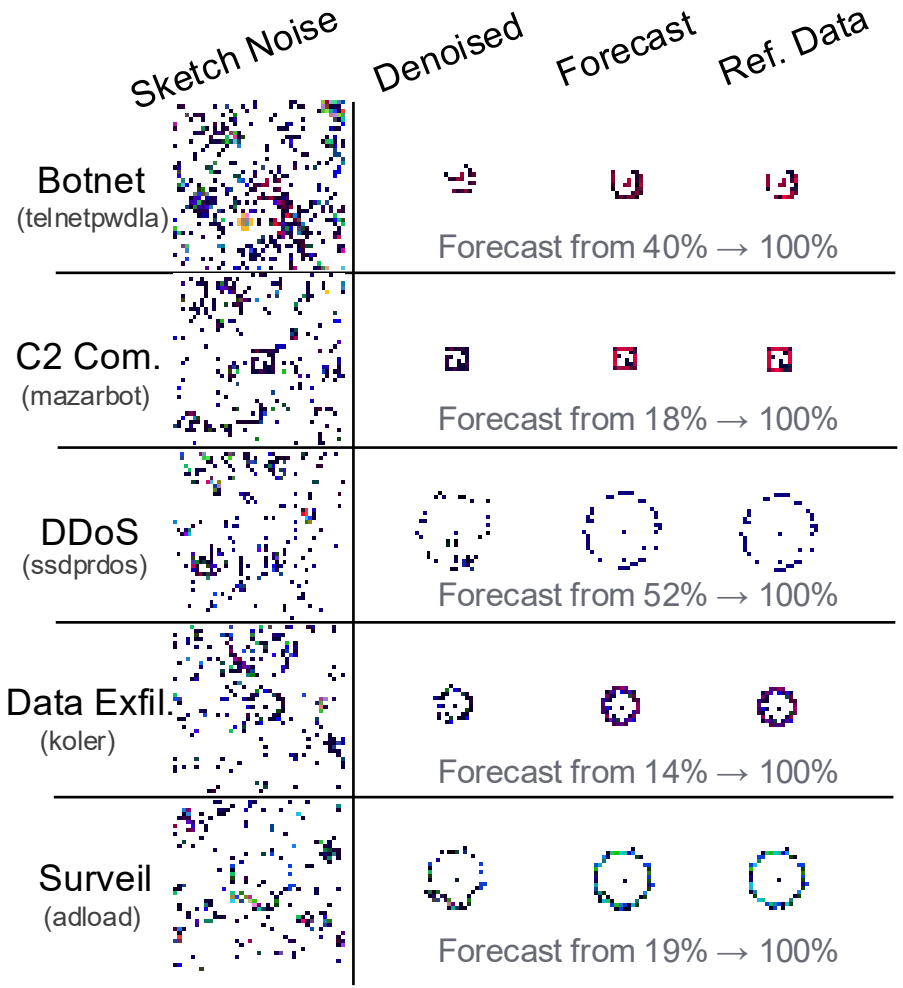
- Comparison with Existing Flow- and Packet-Level ML Detection
  - Classify flow with partial flow observation up to 150 packets
  - Allocate 12 MB memory for bounded memory schemes and relaxed others
- Result
  - Poor performance without per-packet temporal modeling and forecasting capability
  - SketchVision achieves 93% F1-score using only the first 20% of flow observations

Schemes	Feature (Grnlrty)	AUC	ACC	F1	FNR	FPR
nPrintML[1] [108]	Hdr Info. (packet)	0.905	0.985	0.202	0.885	0.000
FlowLens [20]	PS Dist. (flow)	0.970	0.964	0.738	0.420	0.000
SketchFeat. [29]	IPD Dist. (flow)	0.992	0.963	0.695	0.460	0.001
FlowPic [76]	PS&IPD Dist. (flow)	0.999	0.982	0.874	0.220	0.000
SketchVision	PS&IPD (packet)	**0.997**	**0.989**	**0.933**	**0.041**	**0.008**

IPD/PS: Inter-Packet Delay/Packet Size **FlowLens, nPrintML**[1] use RF, others CNN

Visual Examples

(Denoise → Forecast → Classify)



Diffusion-Based Sketch Denoising

(Denoise → Forecast → Classify)

- Per-Flow Painting Performance
  - Baseline (no sketch noise): optimal
  - Sketch noise present: random predication
  - Denoised sketch: restore performance

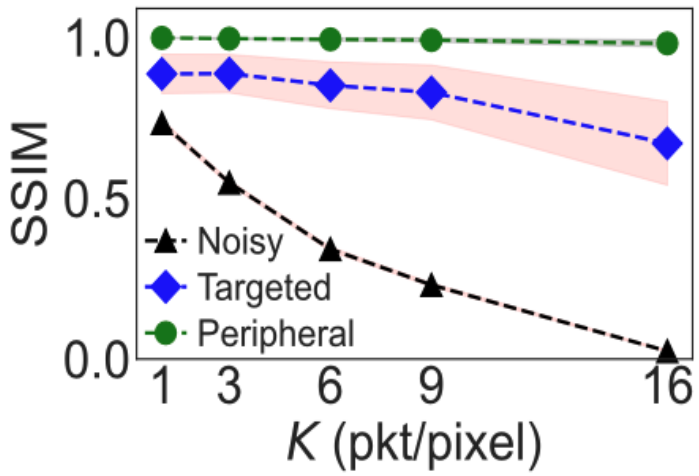
Category	Input Data	AUC	ACC	F1	FNR	FPR
Botnet	Baseline	0.999	0.998	0.976	0.021	0.000
	With noise	0.504	0.978	0.000	1.000	0.000
	Denoised	**0.990**	**0.996**	**0.905**	**0.035**	**0.004**
C2 Com.	Baseline	0.999	0.998	0.980	0.009	0.001
	With noise	0.500	0.976	0.000	1.000	0.000
	Denoised	**0.977**	**0.994**	**0.894**	**0.044**	**0.005**
DDoS	Baseline	1.000	0.999	0.999	0.000	0.000
	With noise	0.519	0.959	0.065	0.963	0.000
	Denoised	**0.998**	**0.997**	**0.968**	**0.002**	**0.003**
Data Exfil.	Baseline	0.999	0.999	0.992	0.003	0.000
	With noise	0.500	0.966	0.000	1.000	0.000
	Denoised	**0.953**	**0.986**	**0.848**	**0.091**	**0.012**
Surveil	Baseline	0.999	0.999	0.991	0.002	0.000
	With noise	0.500	0.975	0.000	1.000	0.000
	Denoised	**0.994**	**0.998**	**0.953**	**0.018**	**0.002**

Diffusion-Based Sketch Denoising

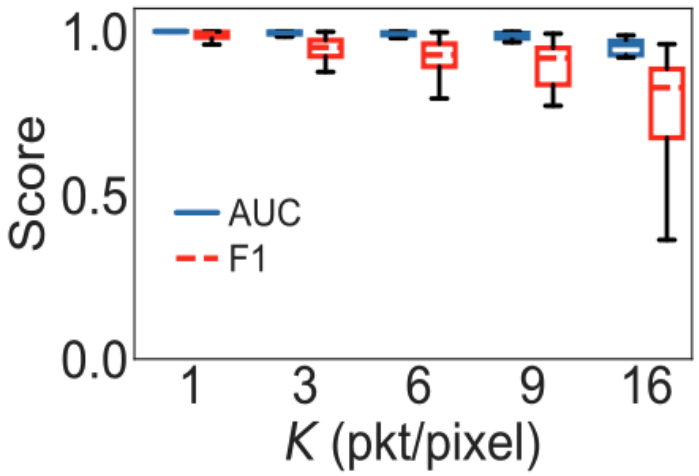
(**Denoise** → Forecast → **Classify**)

- Performance under varying memory configuration
  - K = packet-to-pixel ratio

Category	Input Data	AUC	ACC	F1	FNR	FPR
Botnet	Baseline	0.999	0.998	0.976	0.021	0.000
	With noise	0.504	0.978	0.000	1.000	0.000
	Denoised	**0.990**	**0.996**	**0.905**	**0.035**	**0.004**
C2 Com.	Baseline	0.999	0.998	0.980	0.009	0.001
	With noise	0.500	0.976	0.000	1.000	0.000
	Denoised	**0.977**	**0.994**	**0.894**	**0.044**	**0.005**
DDoS	Baseline	1.000	0.999	0.999	0.000	0.000
	With noise	0.519	0.959	0.065	0.963	0.000
	Denoised	**0.998**	**0.997**	**0.968**	**0.002**	**0.003**
Data Exfil.	Baseline	0.999	0.999	0.992	0.003	0.000
	With noise	0.500	0.966	0.000	1.000	0.000
	Denoised	**0.953**	**0.986**	**0.848**	**0.091**	**0.012**
Surveil	Baseline	0.999	0.999	0.991	0.002	0.000
	With noise	0.500	0.975	0.000	1.000	0.000
	Denoised	**0.994**	**0.998**	**0.953**	**0.018**	**0.002**



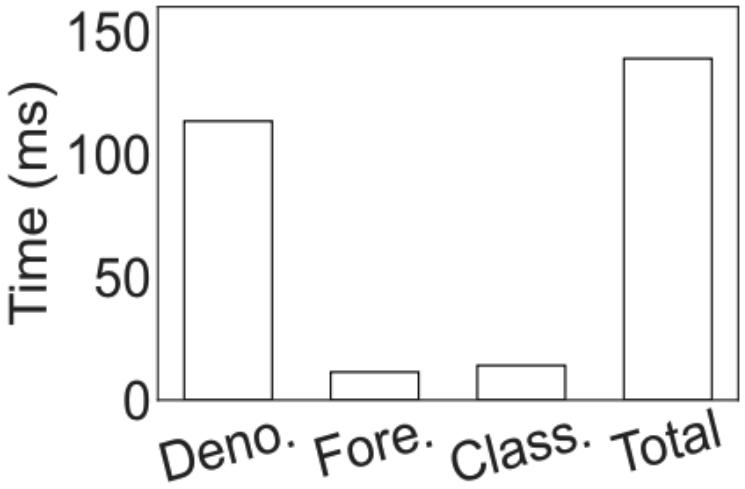
Recovered Sketch Data



ML Performance

Latency & Throughput

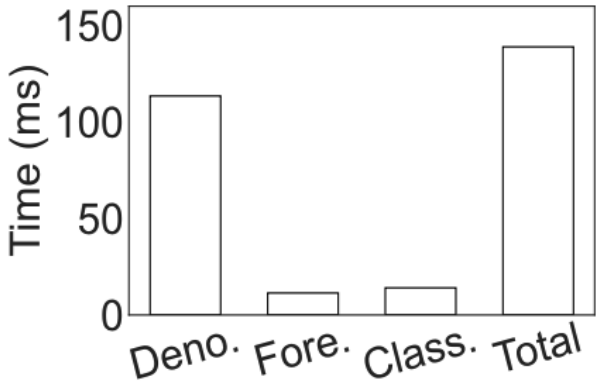
- Data Plane Model
  - Comparable to the state-of-the sketch implementation in eBPF
- ML Model
  - ~140ms total pipeline latency



Per-model Latency

Latency & Throughput

- Data Plane Model
  - Comparable to the state-of-the sketch implementation in eBPF
- ML Model
  - ~140ms total pipeline latency



Per-model Latency

Limitation and Future Work

- Limited Throughput ~ 2K flows/s
  - Uses off-the-shelf diffusion model
- Future Work
  - Accelerate diffusion process & model deployment

Conclusion

- Limitation of existing solution for slow-and-low attack detection
  - Lacking scalable flow behavior monitoring
  - Evading coarse behavior-based detection
  - Partial view of slow-and-low attacks

- **SketchVision**
  - Scalable flow behavior monitoring via sketching
  - Trading computing unit for memory via diffusion model denoising
  - Feasible solution for early detection

- Source code: https://github.com/NIDS-LAB/SketchVision

Q & A

Thank You!