AdaptiveOnline-IDS: Efficient Intrusion Detection via Lifelong Learning and Knowledge Distillation

Nadia Niknami[1] *and Jie Wu*[2]

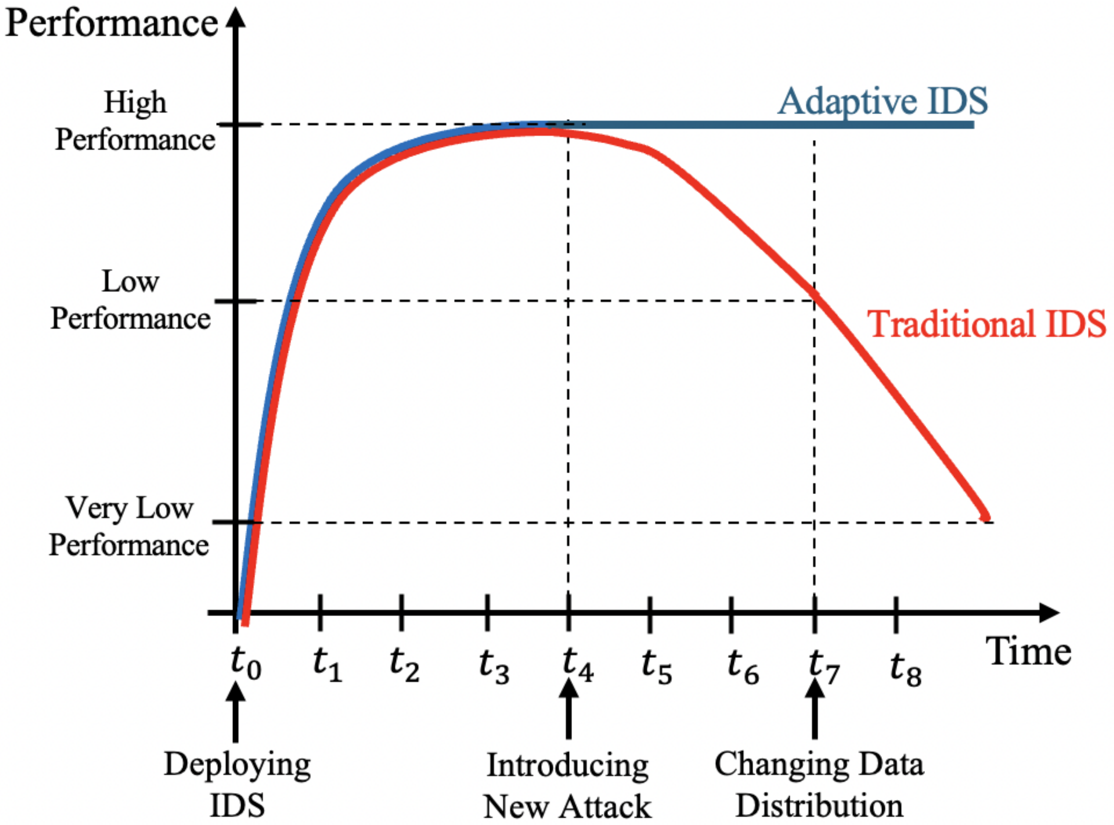
[1] *Villanova University*
[2] *Temple University*

Outline

- Why traditional IDS fails!
- AdaptiveOnline-IDS framework
- Sequential lifelong learning
- Experimental insights

Motivation

- Why do we even need lifelong learning?
- Traditional/batch IDS: fails in evolving environments
- Lifelong-learning IDS: helps maintain performance



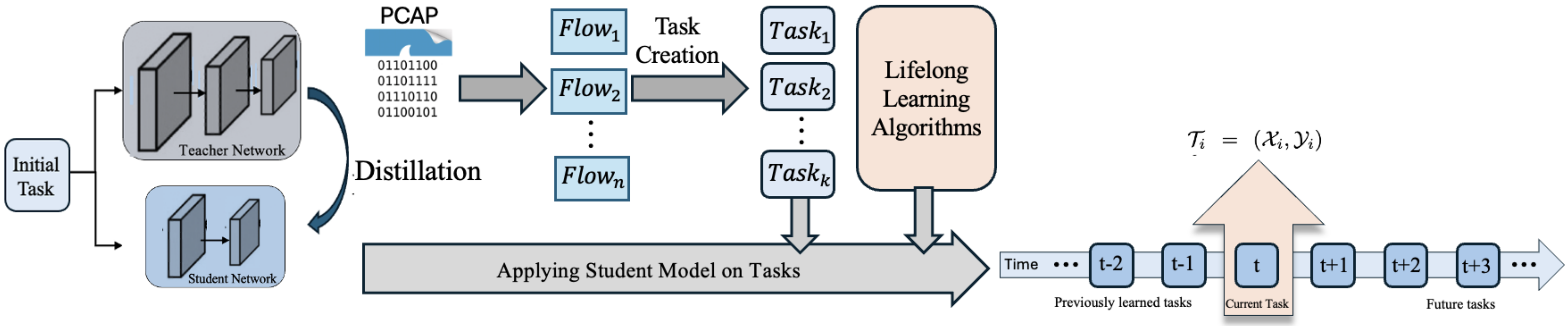
Contribution Summary

- Lifelong Learning for Evolving Attacks
- Knowledge Distillation for Lightweight Deployment
- Sequential Task Adaptation
- Evaluation under Task-order Sensitivity

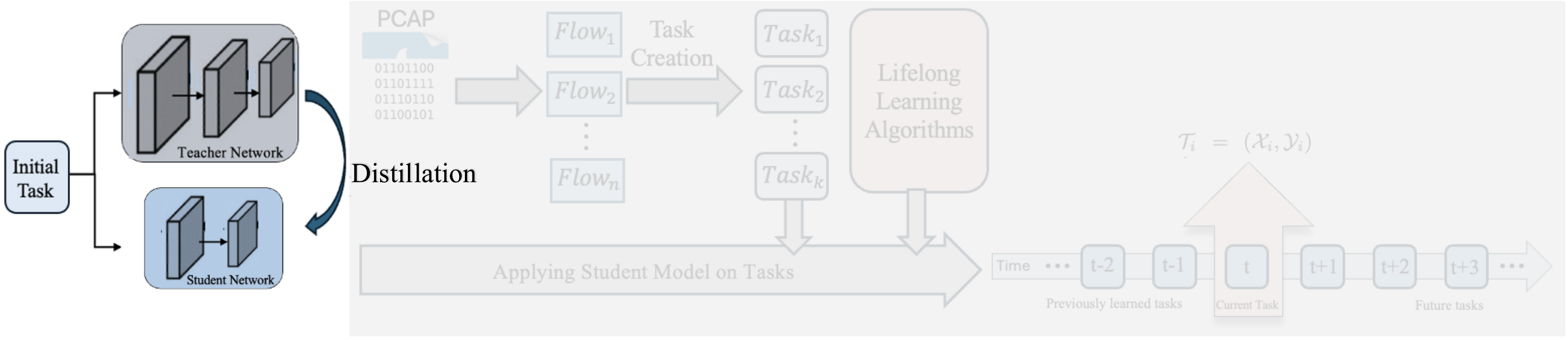
Why AdaptiveOnline-IDS?

Method	Adaptive to New Attacks	Prevents Forgetting	Lightweight Deployment	Sequential Learning
Traditional IDS	✗	✗	✓	✗
KD-based IDS	Limited	✗	✓	✗
LL-based IDS	✓	Partial	✗	✓
AdaptiveOnline-IDS	✓	✓	✓	✓

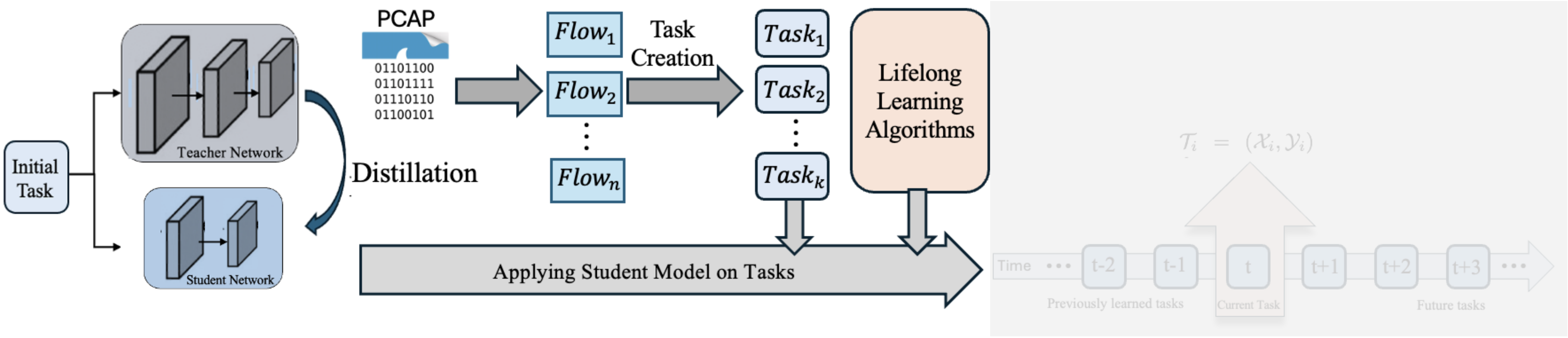
Idea Overview



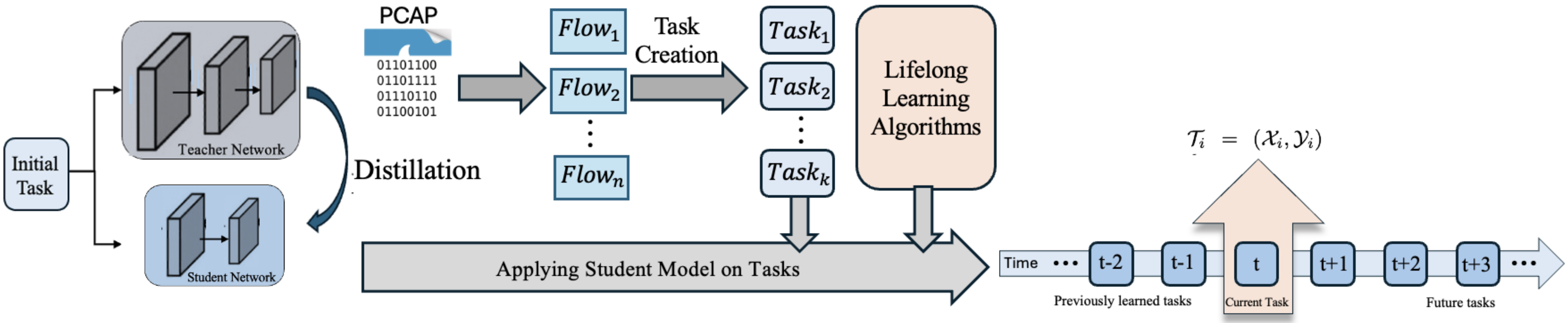
Idea Overview (Step 1: Distillation)



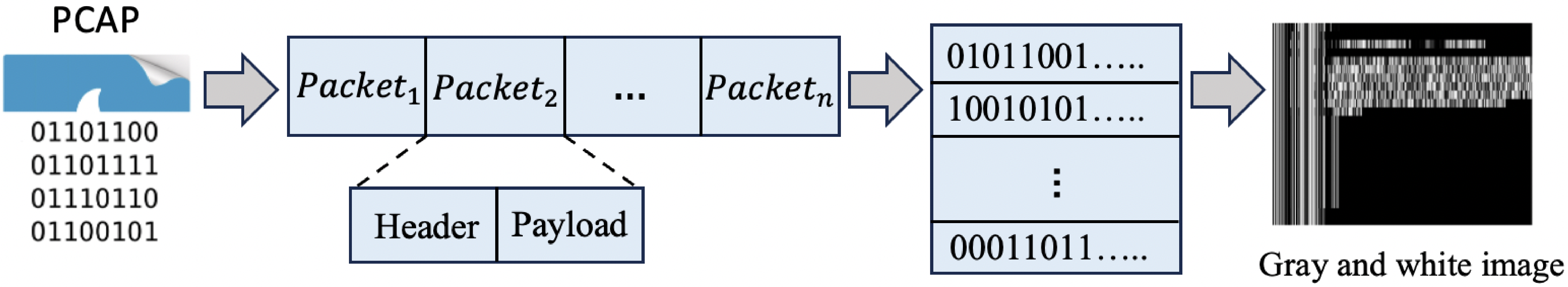
Idea Overview (Step 2: PCAP flows → tasks)



Idea Overview (Step 3: LL Updating)

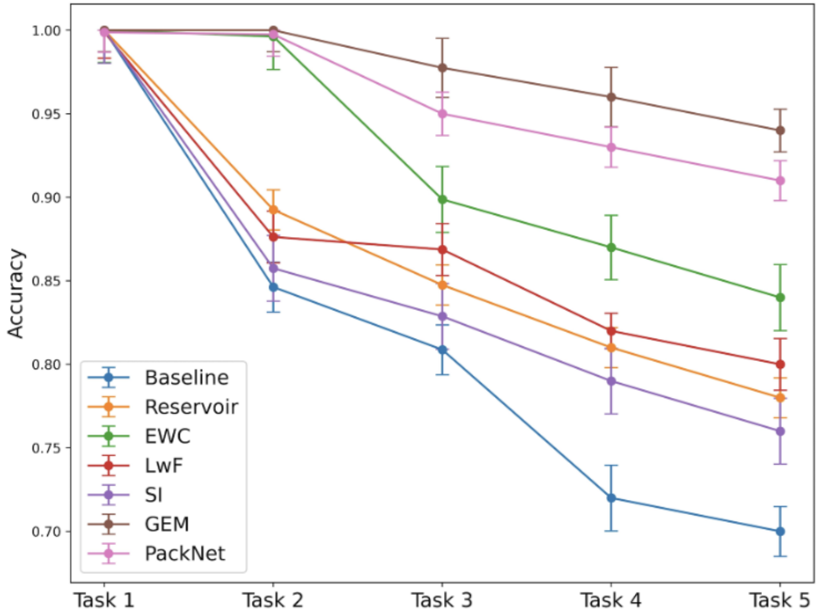


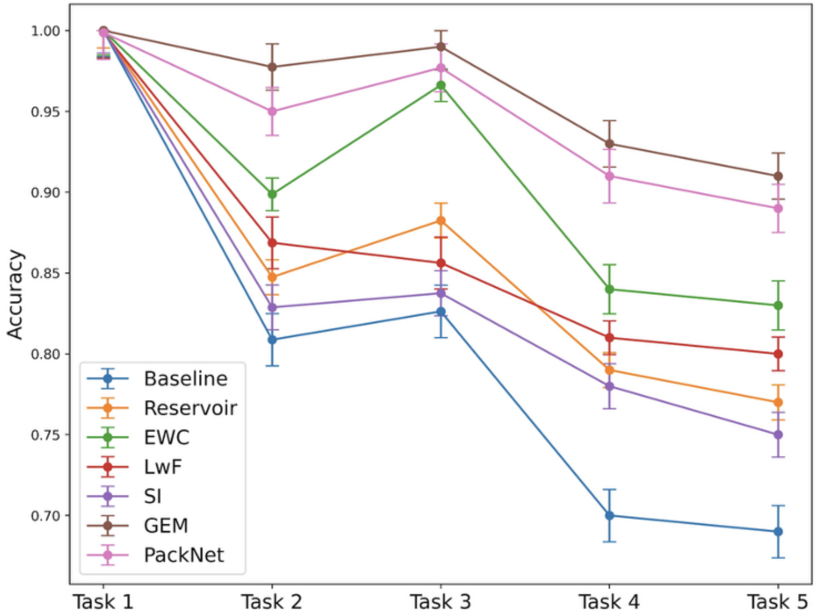
PCAP-to-image Conversion Process



Lifelong Learning Algorithms

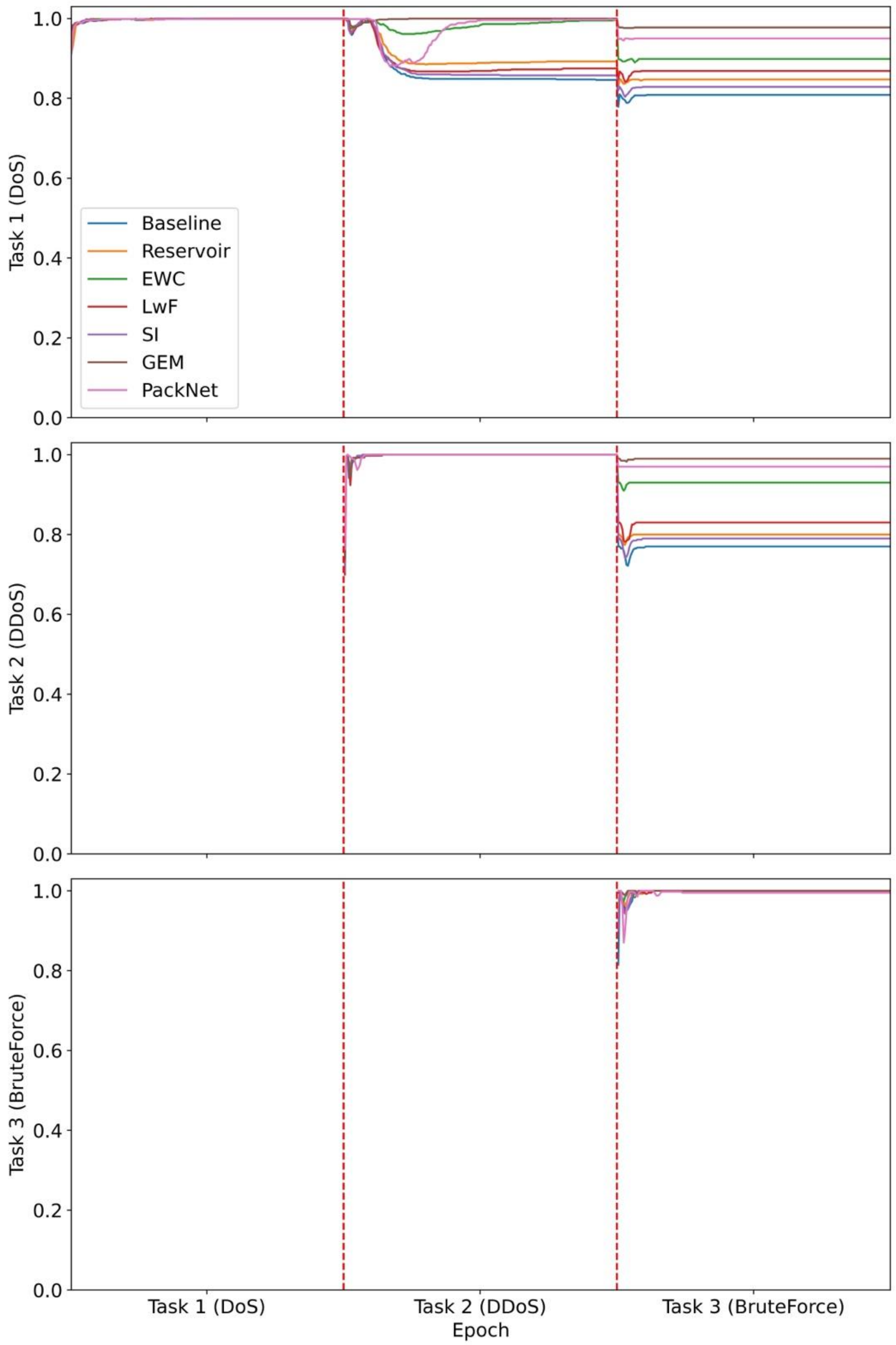
- Accuracy decreases after task transitions due to catastrophic forgetting.
- Similar attack tasks produce smaller performance degradation.
- GEM and PackNet maintain stronger stability during sequential learning.
- Performance retention is critical for adaptive IDS deployment





Accuracy on Across Tasks

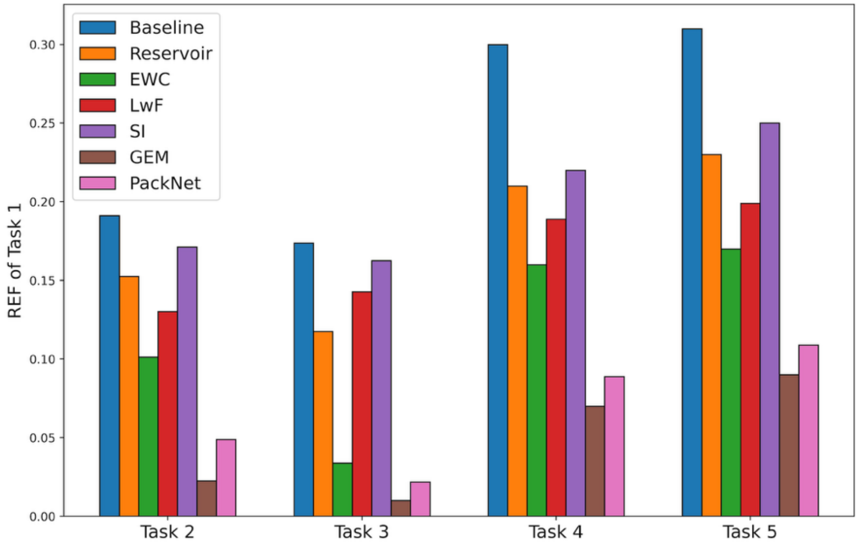
- Sequential learning introduces catastrophic forgetting.
- GEM and PackNet maintain the highest stability across tasks.
- Performance degradation increases when task similarity decreases.
- Lifelong learning enables continuous adaptation without full retraining

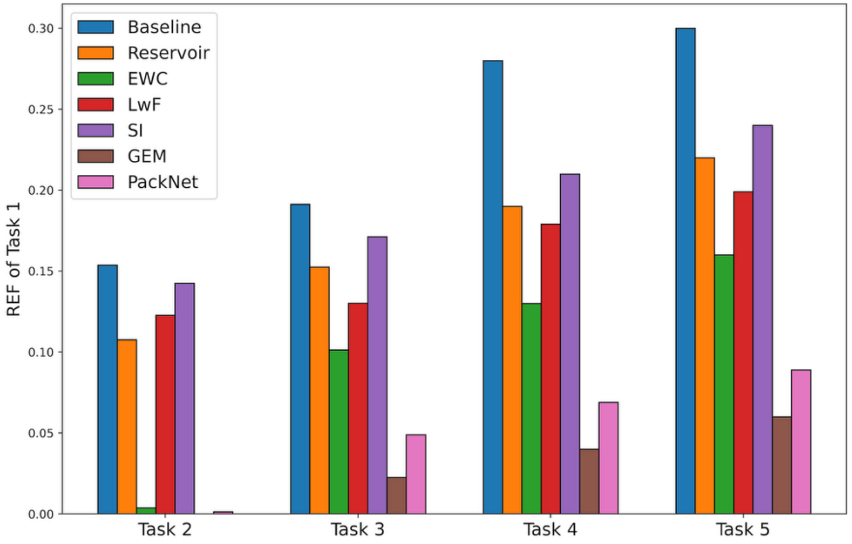


Relative Experience Forgetting (REF)

$$\mathrm{REF}(t_i, t_j) = (a_{t_i} - a_{t_i}^{t_j})/a_{t_i},$$

- How much a model forgets earlier tasks once it has been trained on new ones.
- The relative drop in accuracy after subsequent training on tasks





KD Table

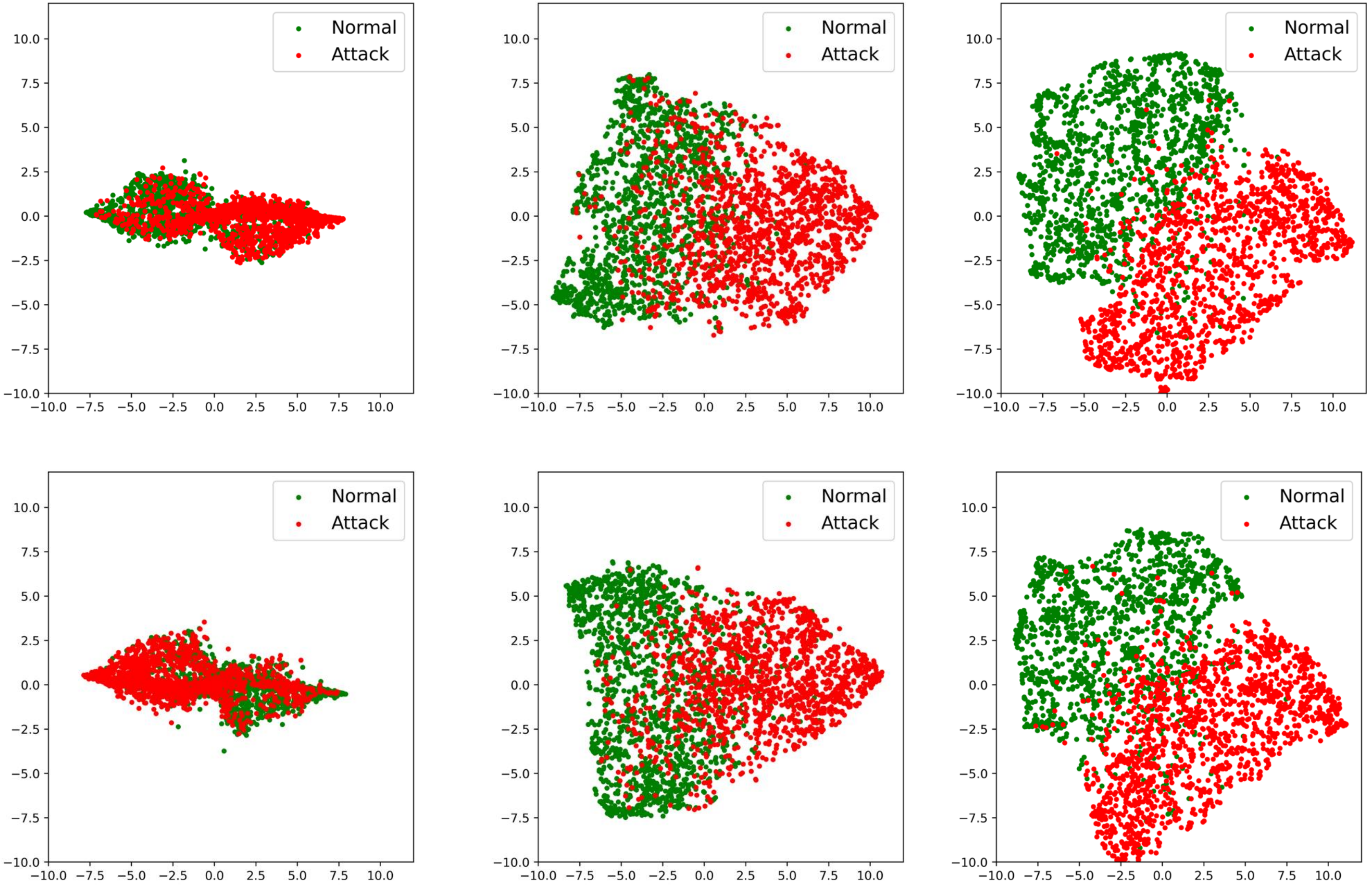
TABLE I
PERFORMANCE COMPARISON OF TEACHER AND STUDENT KD MODELS ACROSS KEY METRICS.

Model	Training Time (s)	Memory Usage (MB)	Inference Time (s)	Accuracy
Teacher (5-layer CNN)	91.07	8.66	0.314	0.95
Student (4-layer CNN)	74.09	0.95	0.254	0.94
Student (3-layer CNN)	65.99	0.33	0.195	0.93
Student (2-layer CNN)	50.35	0.13	0.117	0.93

- KD enables efficient lightweight IDS deployment.
- Student models preserve near-teacher performance.
- Memory usage and inference time decrease substantially.
- The 4-layer Student provides the best accuracy-efficiency tradeoff

t-SNE

- Feature separation improves as new tasks are learned sequentially.
- Attack and normal traffic become more distinguishable over time.
- GEM preserves previous knowledge while adapting to new attacks



Conclusion & Future Work

- Adaptive IDS must continuously evolve
- KD enables practical lightweight deployment
- GEM and PackNet best preserve prior knowledge
- Task order significantly impacts continual learning