ICLR'21 Spotlight Talk

A Panda? No, It's Sloth: Slowdown Attacks on Adaptive Multi-Exit Neural Network Inference

Sanghyun Hong*, **Yiğitcan Kaya***, Ionuţ-Vlad Modoranu, Tudor Dumitraș

University of Maryland, College Park

MARYLAND CYBERSECURITY CENTER

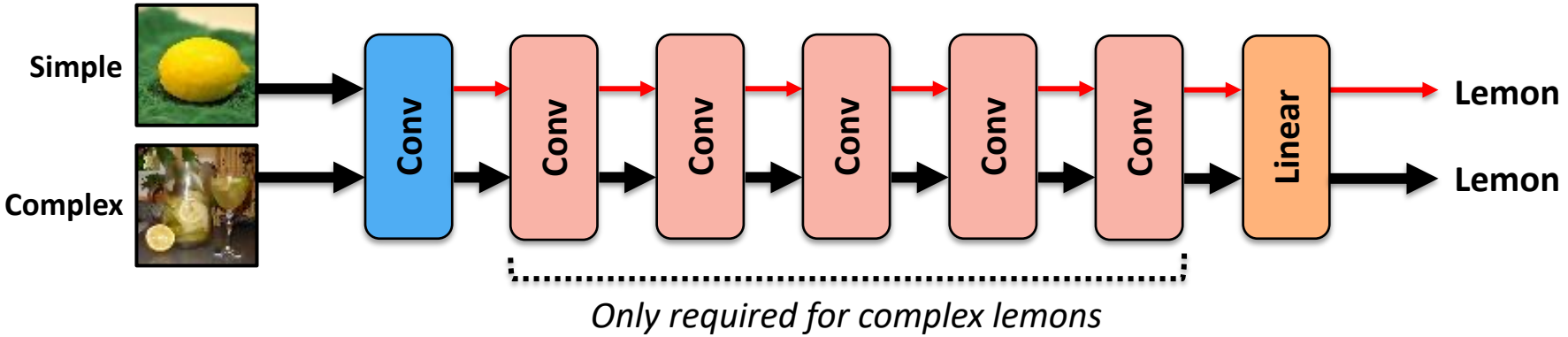
Deeper Networks (Generally) Have Better Performances

- **On ImageNet**
  - ResNet[1] **improves top-1 acc. by 19%** over AlexNet
  - EfficientNet[2] **improves top-1 acc. by 10%** over ResNet

[1]He et al., *Deep Residual Learning for Image Classification*, CVPR'16
[2]Tan et al., *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*, ICLR'19

Deeper Networks *Overthink*[1]

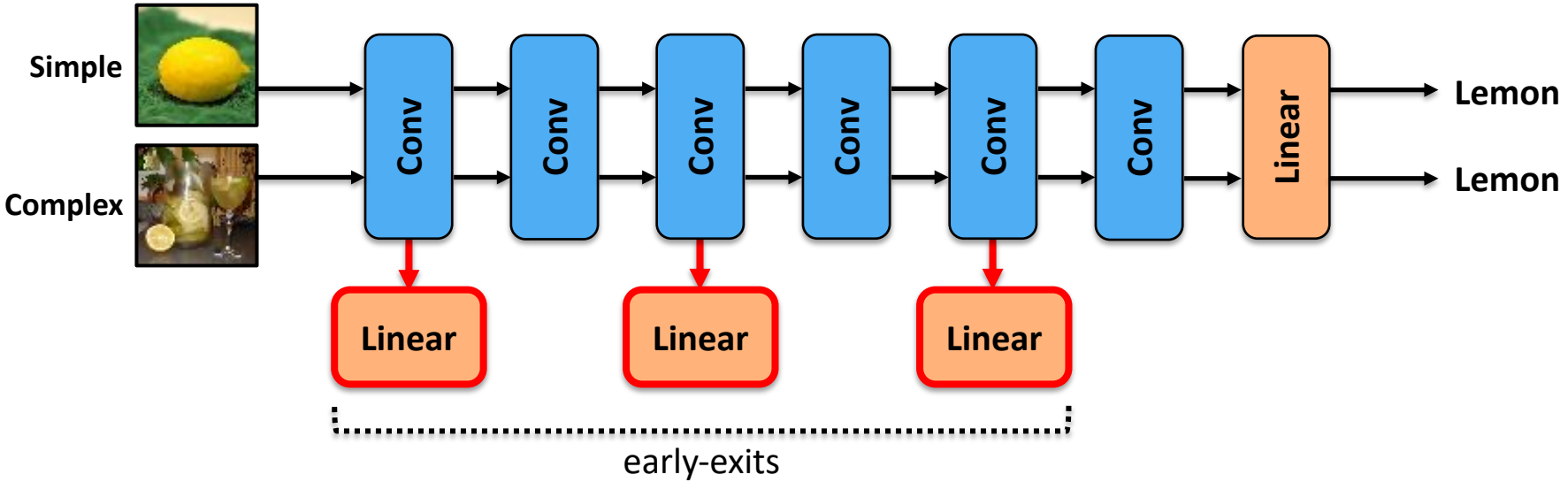
- The networks spend the same amount of computations for every input



[1]Kaya et al., *Shallow-Deep Networks,* ICML'19

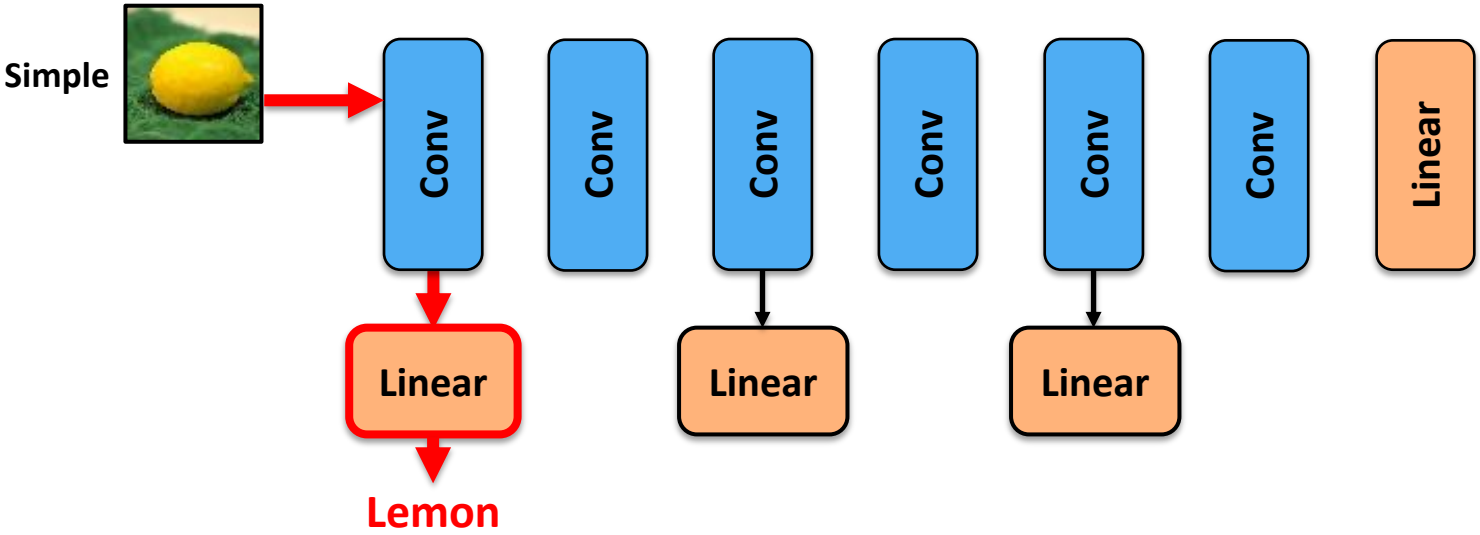
Multi-exit Neural Network Architectures

- Key idea: ***Input-adaptive*** inference



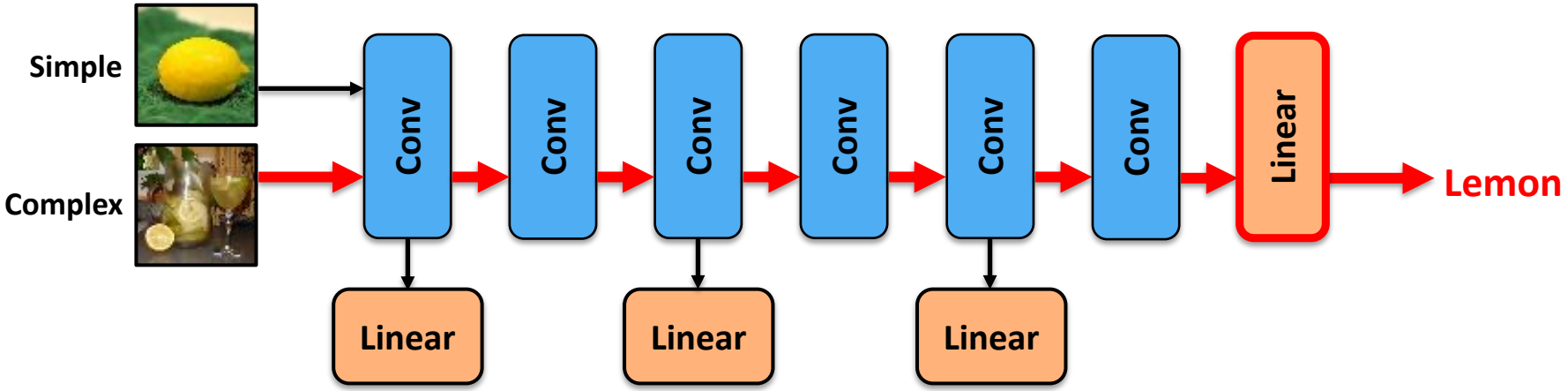
Input-adaptive Inference

- **Simple inputs (~80% of CIFAR-10)** are correctly classified at earlier-exits



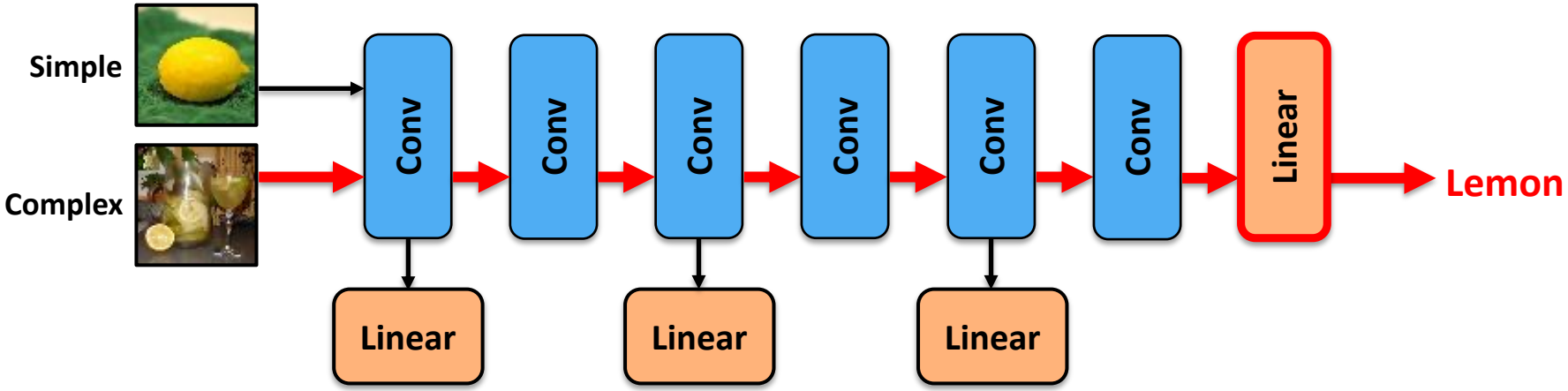
Input-adaptive Inference

- **Complex inputs (< 10% of CIFAR-10)** only harness its deeper layers



Input-adaptive Inference

- **Complex inputs (< 10% of CIFAR-10)** only harness its deeper layers



During inference, they can save computations ~50% while preserving accuracy

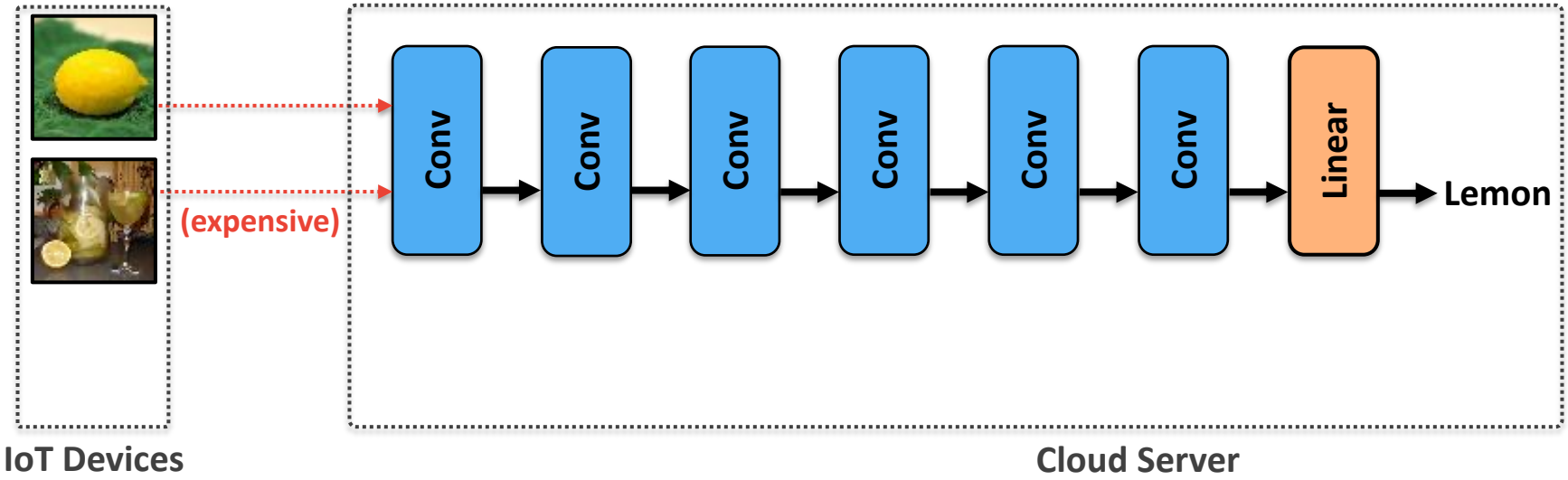
Multi-exit Models Receive Attentions from the Community

- In NLP, BERT[12] uses multi-exits to enable fast and efficient inference

[1]Xin al., *DeeBERT: Dynamic Early Exiting for Accelerating BERT Inference*, ACL'20
[2]Liu et al., *FastBERT: a Self-distilling BERT with Adaptive Inference Time*, ACL'20

Multi-exit Models Receive Attentions from the Community

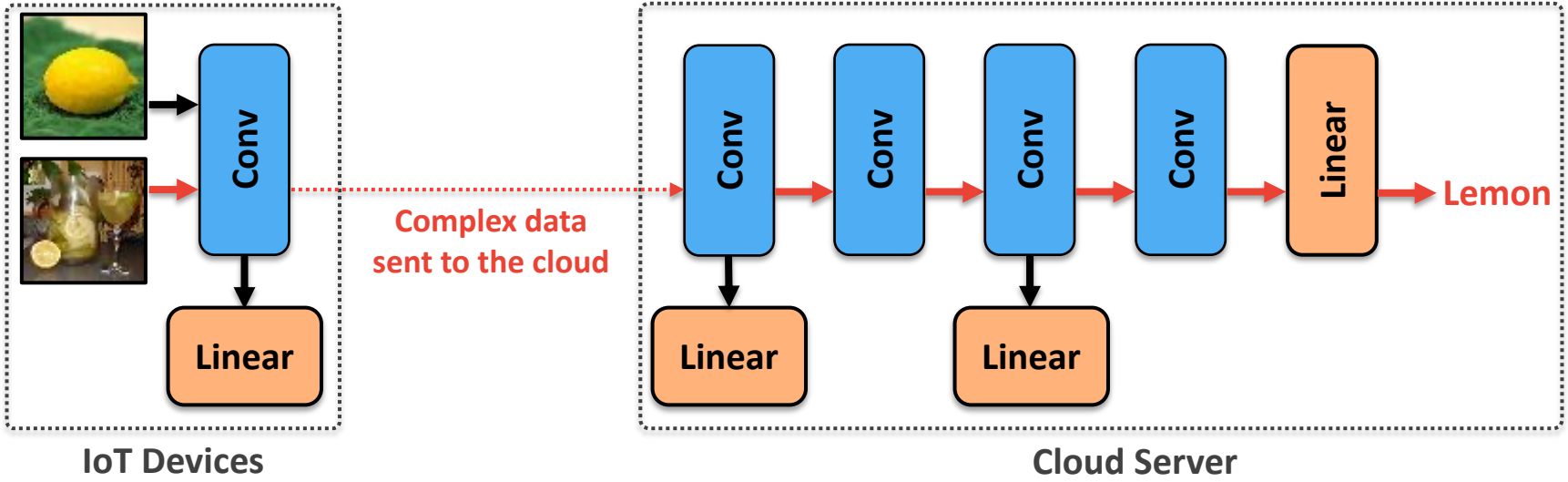
- In NLP, BERT uses multi-exits to enable fast and efficient inference
- Prior work[123] partitions multi-exit networks to deploy DNNs to IoT devices



[1]Kang al., *Neurosurgeon: Collaborative intelligence between the cloud and mobile edge*, ASPLOS'17
[2]Li et al., *On-demand deep learning model co-inference with device-edge synergy*, MECOMM'18
[3]Zhou et al., *Distributing deep neural networks with containerized partitions at the edge*, HotEdge'20

Multi-exit Models Receive Attentions from the Community

- In NLP, BERT uses multi-exits to enable fast and efficient inference
- Prior work[123] partitions multi-exit networks to deploy DNNs to IoT devices



[1]Kang al., *Neurosurgeon: Collaborative intelligence between the cloud and mobile edge*, ASPLOS'17
[2]Li et al., *On-demand deep learning model co-inference with device-edge synergy*, MECOMM'18
[3]Zhou et al., *Distributing deep neural networks with containerized partitions at the edge*, HotEdge'20

Are the Computational Benefits Provided by the Input-adaptive Multi-exit NNs Robust against Adversarial Input Perturbations?

Standard Adversarial Examples Do Not Cause Delays

- Existing evasion attacks **fail to reduce efficacy**
  - **Multi-exit models** : 3 SDNs (VGG16-, ResNet56-, MobileNet-based)
  - **Evasion attacks**(l_∞) : PGD[1], UAP[2], PGD-avg. and PGD-max.[3]

[1]Madry et al., *Towards Deep Learning Models Resistant to Adversarial Attacks*, ICLR'18
[2]Moosavi-Dezfooli et al., *Universal Adversarial Perturbations*, CVPR'17
[3]Hu et al., *Triple wins: Boosting accuracy, robustness, and efficiency together by enabling input-adaptive inference*, ICLR'20

We Provide Tools To Study This New Threat *Systematically*

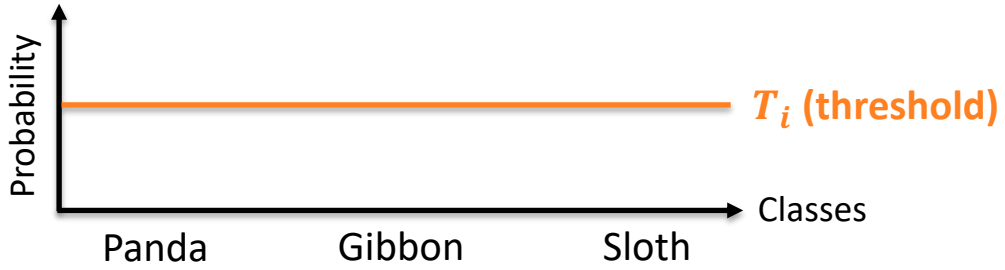
- **DeepSloth:** a PGD-like attack against multi-exit models
- **Efficacy:** a standardized metric to compare slowdowns

We make our implementation public at GitHub!

Practical Stopping Criteria in Multi-exit Neural Networks

- T_i: A threshold that determines whether to take an exit i or not
  - *Confidence-based* (SDN[1]) / *entropy-based* (MSDNet[2])

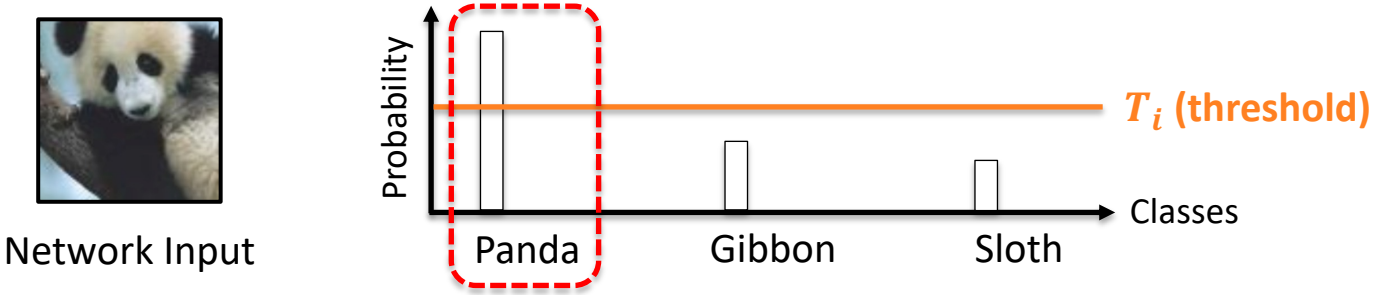
Network Input



[1]Kaya et al., *Shallow-Deep Networks*, ICML'19
[2]Huang et al., *Multi-Scale Dense Networks for Resource Efficient Image Classification*, ICLR'18

Practical Stopping Criteria in Multi-exit Neural Networks

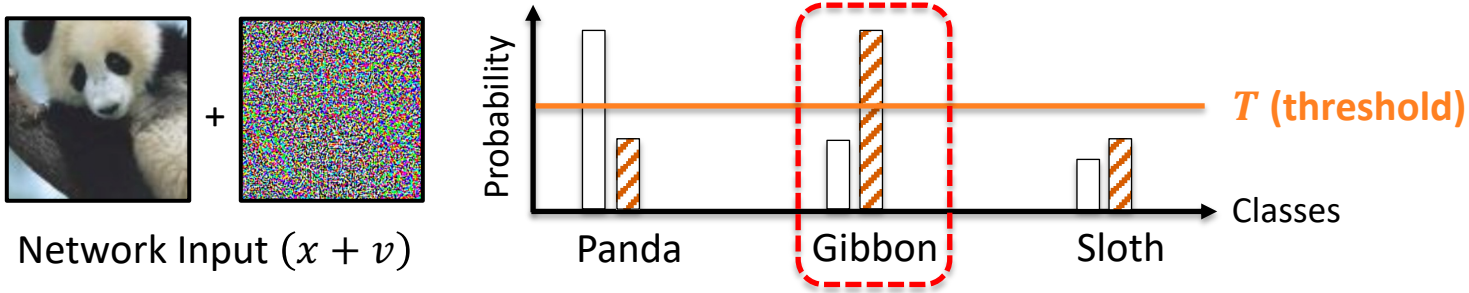
- T_i: A threshold that determines whether to take an exit i or not
  - *Confidence-based* (SDN[1]) / *entropy-based* (MSDNet[2])



[1]Kaya et al., *Shallow-Deep Networks*, ICML'19
[2]Huang et al., *Multi-Scale Dense Networks for Resource Efficient Image Classification*, ICLR'18

Standard Adversarial Examples Do Not Cause Delays

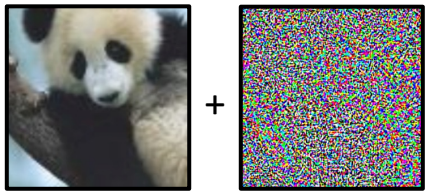
- T_i: A threshold that determines whether to take an exit i or not
  - *Confidence-based* (SDN) / *entropy-based* (MSDNet)



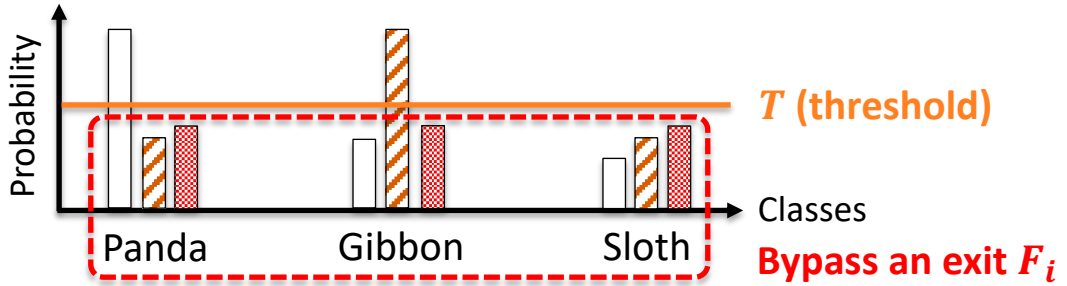
- **PGD attack's objective:** $v^{t+1} = \prod_{||v||<\epsilon} \left(v^t + \alpha \, \mathrm{sgn}\left(\nabla_v \mathcal{L}(\theta, F(x+v), y)\right)\right)$

DeepSloth: Push F_i Towards the Uniform $\overline{y}$

- T_i: A threshold that determines whether to take an exit i or not
  - *Confidence-based* (SDN) / *entropy-based* (MSDNet)



Network Input $(x + v)$

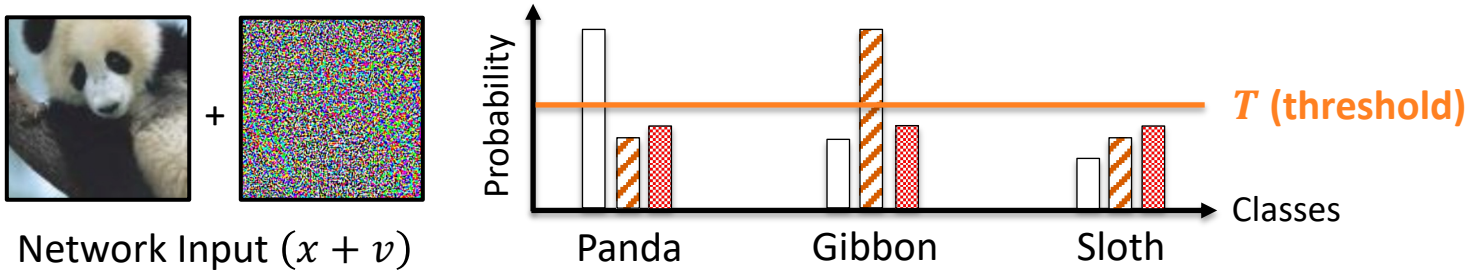


- **DeepSloth's objective:** $v^{t+1} = \prod_{||v||<\epsilon} \left(v^t + \alpha \, \mathrm{sgn}\left(\nabla_v \mathcal{L}(\theta, F_i(x+v), \overline{y}) \right) \right)$

$\overline{y}$: uniform dist.

DeepSloth: Consider All Exit-points F_i

- T_i: A threshold that determines whether to take an exit i or not
  - *Confidence-based* (SDN) / *entropy-based* (MSDNet)

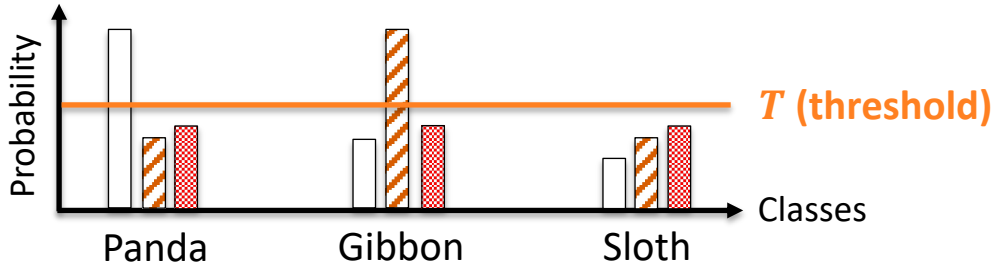


- **DeepSloth's objective:** $v^{t+1} = \prod_{||v||<\epsilon} \left(v^t + \alpha \, \mathrm{sgn} \left(\nabla_v \sum_{x \in D'} \sum_{0<i<K} \mathcal{L}(\theta, F_i(x+v), \overline{\boldsymbol{y}}) \right) \right)$

DeepSloth: Bypass All Exit-points

- T_i: A threshold that determines whether to take an exit i or not
  - *Confidence-based* (SDN) / *entropy-based* (MSDNet)

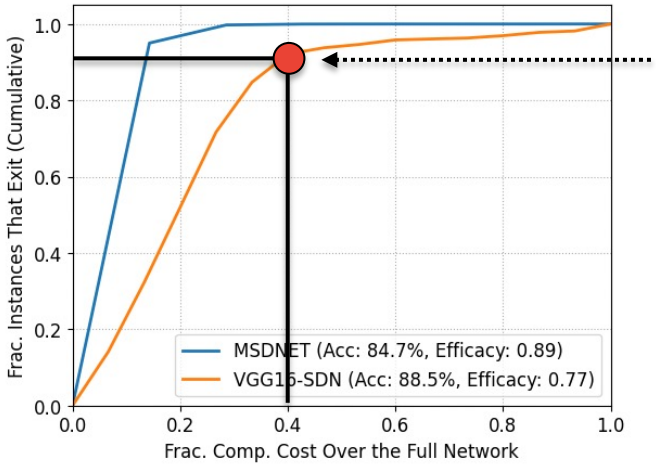
Network Input $(x + v)$



- **DeepSloth's objective:** $v^{t+1} = \prod_{||v||<\epsilon} \left(v^t + \alpha \operatorname{sgn} \left(\nabla_v \sum_{x \in D'} \sum_{0<i<K} \mathcal{L}(\theta, F_i(x+v), \overline{y}) \right) \right)$

Our Metric to Measure The Impact of DeepSloth

- **Efficacy**: area under the early-exit capacity (EEC) curve



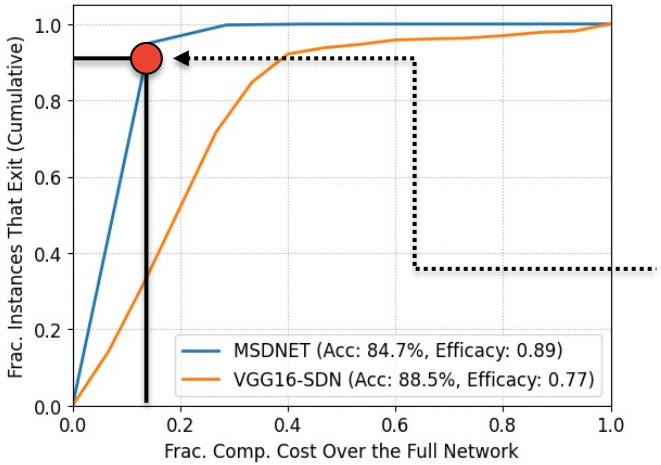
VGG16-SDN

< 40% of computations over a full network

90% of test-time samples will be classified

Our Metric to Measure The Impact of DeepSloth

- **Efficacy**: area under the early-exit capacity (EEC) curve



VGG16-SDN

< 40% of computations over a full network

90% of test-time samples will be classified

MSDNet

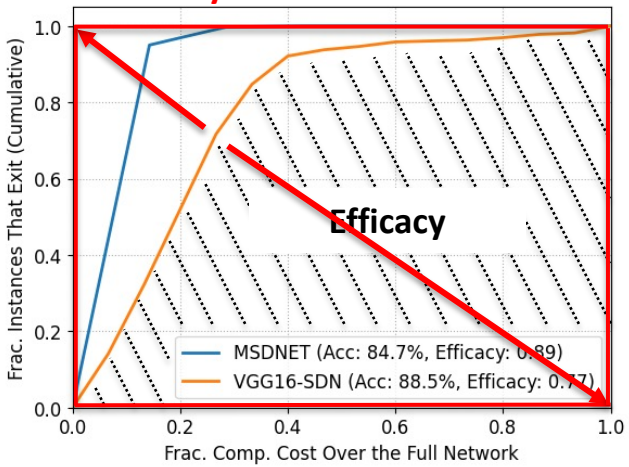
< 18% of computations over a full network

90% of test-time samples will be classified

Our Metric to Measure The Impact of DeepSloth

- **Efficacy**: area under the early-exit capacity (EEC) curve

1.0: Extremely efficient



0.0: No efficacy

VGG16-SDN

< 40% of computations over a full network

90% of test-time samples will be classified

MSDNet

< 18% of computations over a full network

90% of test-time samples will be classified

We can compare efficacy (or slowdowns) across different multi-exit mechanisms

Evaluation: Setup

- **Datasets:** CIFAR-10 and Tiny ImageNet
- **Multi-exit NNs:**
  - Shallow-Deep Networks (SDNs)[1]
  - MSDNets[2]

[1]Kaya et al., *Shallow-Deep Networks*, ICML'19
[2]Huang et al., *Multi-Scale Dense Networks for Resource Efficient Image Classification*, ICLR'18

Effectiveness of DeepSloth in White-Box Scenarios

- **DeepSloth**
  - Multi-exit models show **~0 efficacy** on the samples perturbed by DeepSloth
  - DeepSloth also hurts the multi-exit models' accuracy by **4~99%**

Effectiveness of DeepSloth in White-Box Scenarios

- **DeepSloth**
  - Multi-exit models show **~0 efficacy** on the samples perturbed by DeepSloth
  - DeepSloth also hurts the multi-exit models' accuracy by **4~99%**

- **Universal DeepSloth**
  - **Universal** : a perturbation reduces the efficacy **~15%** for all inputs
  - **Class-universal** : a perturbation reduces the efficacy **~45%** for all inputs in a class

Transferability: Towards Practical Black-Box DeepSloth

- **DeepSloth is transferable**
  [Let's talk more in our poster session]
  1) Cross-architecture
  2) Cross-domain
  3) Cross-mechanism

[1]Wang al., *SkipNet: Learning Dynamic Routing in Convolutional Networks*, ECCV'18

Transferability: Towards Practical Black-Box DeepSloth

- **DeepSloth is transferable**
  [Let's talk more in our poster session]
  1) Cross-architecture
  2) Cross-domain
  3) Cross-mechanism

The Vulnerability of Multi-exit Models to DeepSloth Is Prevalent

[1]Wang al., *SkipNet: Learning Dynamic Routing in Convolutional Networks*, ECCV'18

Real-world Implications: an IoT Deployment Scenario

- **Partition strategy**
  - IoT devices : the first-exit of MSDNet
  - Cloud servers (AWS) : the rest of the multi-exit model

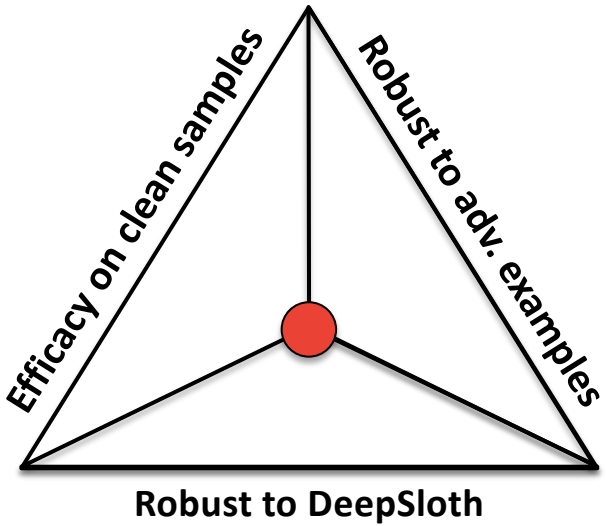
Real-world Implications: an IoT Deployment Scenario

- **Partition strategy**
  - IoT devices : the first-exit of MSDNet
  - Cloud servers (AWS) : the rest of the multi-exit model
- **Latency comparison (in CIFAR-10)**
  - Default latency : **11ms**∼ (all samples are sent to the cloud)
  - Average-case latency : **0.5ms**∼ (reduced **20x** in the partitioned MSDNet)

Real-world Implications: an IoT Deployment Scenario

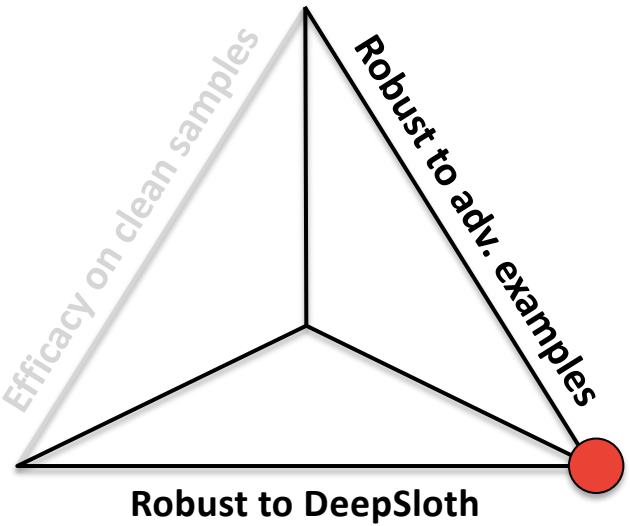
- **Partition strategy**
  - IoT devices : the first-exit of MSDNet
  - Cloud servers (AWS) : the rest of the multi-exit model
- **Latency comparison (in CIFAR-10)**
  - Default latency : **11ms**∼ (all samples are sent to the cloud)
  - Average-case latency : **0.5ms**∼ (reduced **20x** in the partitioned MSDNet)
  - **Worst-case latency** : **11ms**∼ (caused by DeepSloth)

We Observe A Tradeoff Between the Robustness and Efficacy



- Standard adversarial training (AT) is **not** a countermeasure to DeepSloth

We Observe A Tradeoff Between the Robustness and Efficacy



- **An adaptation of AT** with DeepSloth **loses the robustness** against PGD attacks

Conclusion

- Input-adaptive DNNs are not robust against adversarial slowdowns
  - Our attack can **completely offset** the computational savings
  - The attacker can **increase the latency** in an IoT scenario

- We provides tools to study the vulnerability *systematically*
  - **DeepSloth**: a PGD-like attack that crafts adv. examples for causing slowdowns
  - **Efficacy**: a metric to compare slowdowns across different mechanisms

- Standard adversarial training is not a countermeasure
  - **A tradeoff:** { being robust to adv. examples } vs. { being robust to DeepSloth }
  - **Future work:** we need to consider out-of-box defenses to mitigate this threat

Thank You!

Sanghyun Hong [shhong@cs.umd.edu]

Don’t Be A Sloth! Please Come to Our Poster Session ☺