Not all Textual Instances are Alike: Efficient NLP by Better Understanding of our Data

Roy Schwartz

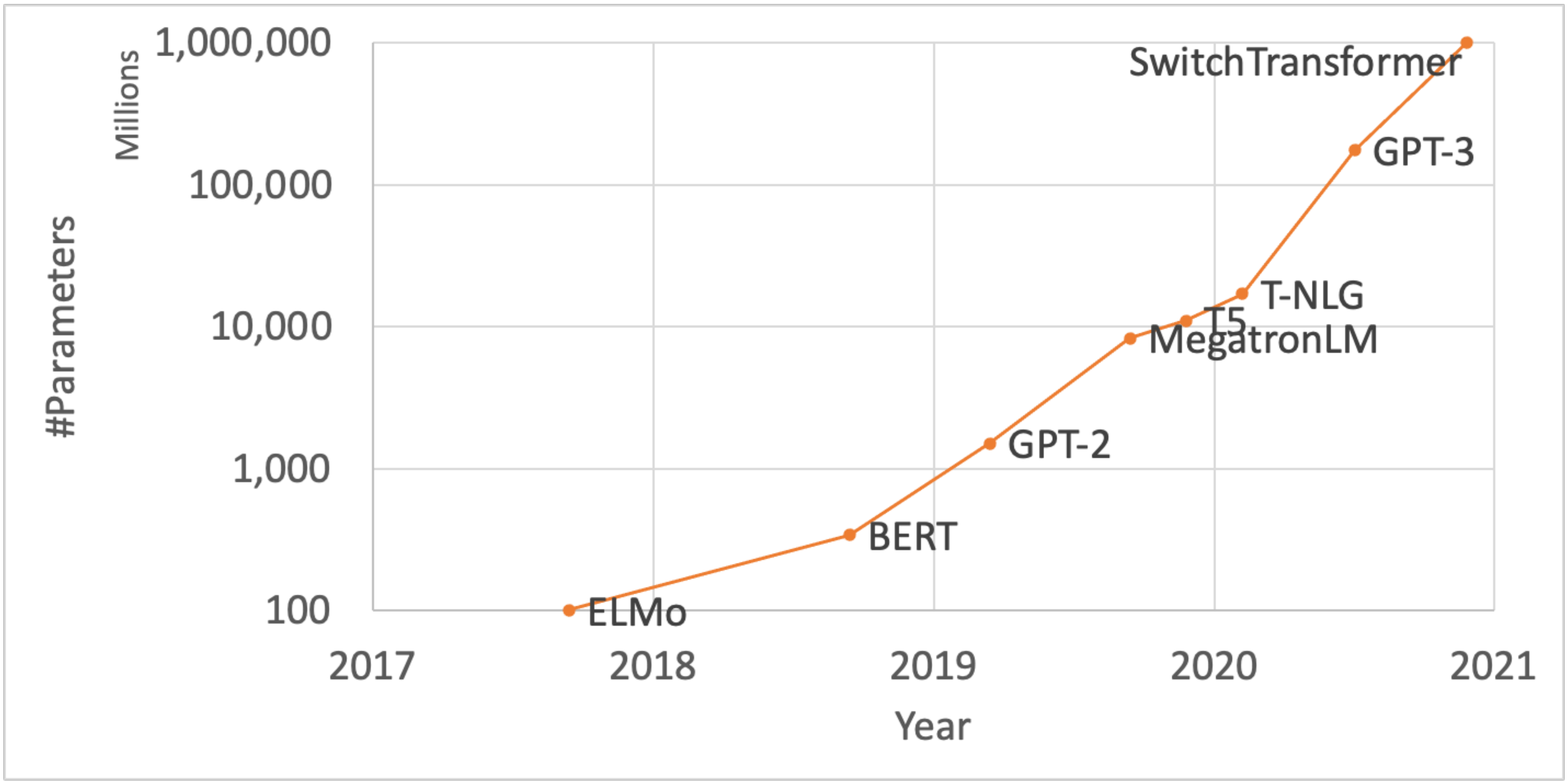
Hebrew University of Jerusalem

SustainNLP 2021

THE HEBREW UNIVERSITY OF JERUSALEM

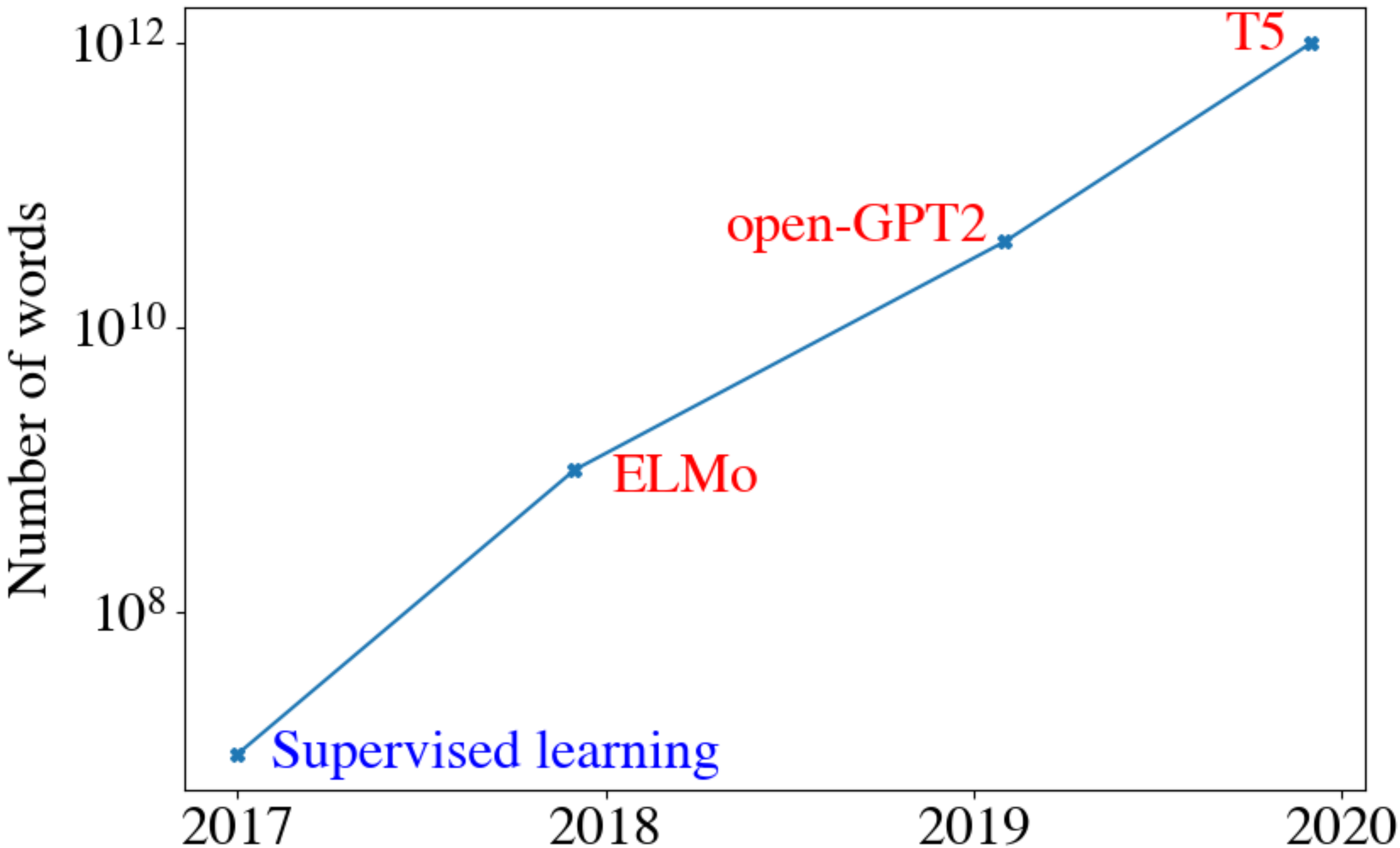
Premise: Big Models

10,000X in 3 Years



Large Datasets

100,000X in 3 Years



Efficiency

Current Approaches

Efficiency

Current Approaches

Efficiency

Current Approaches

- **Model distillation**
  - Hinton et al. (2015); MobileBERT (Sun et al., 2019); DistilBERT (Sanh et al., 2019)
- **Pruning / Structural Pruning**
  - Han et al. (2016); SNIP (Lee et al., 2019); LTH (Frankle & Corbin, 2019); MorphNet (Gordon et al., 2018); Michel et al. (2019); LayerDrop (Fan et al., 2020); Dodge, **Schwartz** et al. (2019)
- **Quantization**
  - Gong et al. (2014); Q8BERT (Zafrir et al., 2019); Q-BERT (Shen et al., 2019)

Data in NLP

Basic Assumption: Instances are IID

Not all Instances are Alike

1. *The movie was awesome.*
2. *I could definitely see why this movie received such great critiques, but at the same time I can't help but wonder whether the plot was written by a 12 year-old or by an award-winning writer.*

Not all Instances are Alike

1. *The movie was awesome.*
2. *I could definitely see why this movie received such great critiques, but at the same time I can't help but wonder whether the plot was written by a 12 year-old or by an award-winning writer.*

What is the capital of Italy?	**Which country won the largest number of swimming medals in the 2016 summer olympics?**	**Would a glass of water that falls from 10 feet down to a trampoline break?**

Outline

Not all Instances are Alike

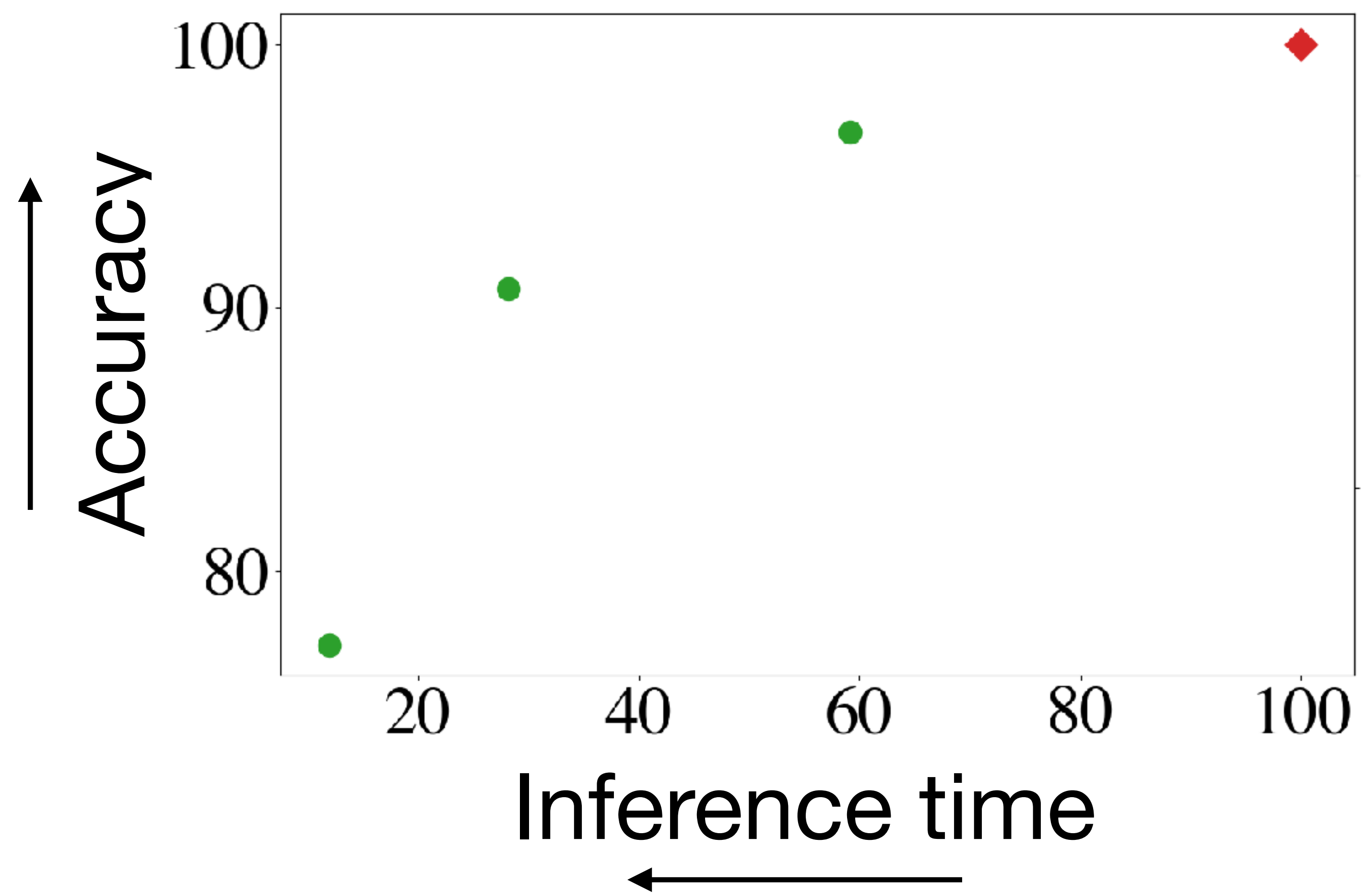
- Efficient inference
    - Schwartz et al., ACL 2020
- Efficient training
    - Swayamdipta et al., EMNLP 2020
- Better masked language modeling for vision and language
    - Bitton et al., Findings of EMNLP 2021

Case Study 1: Efficient Inference

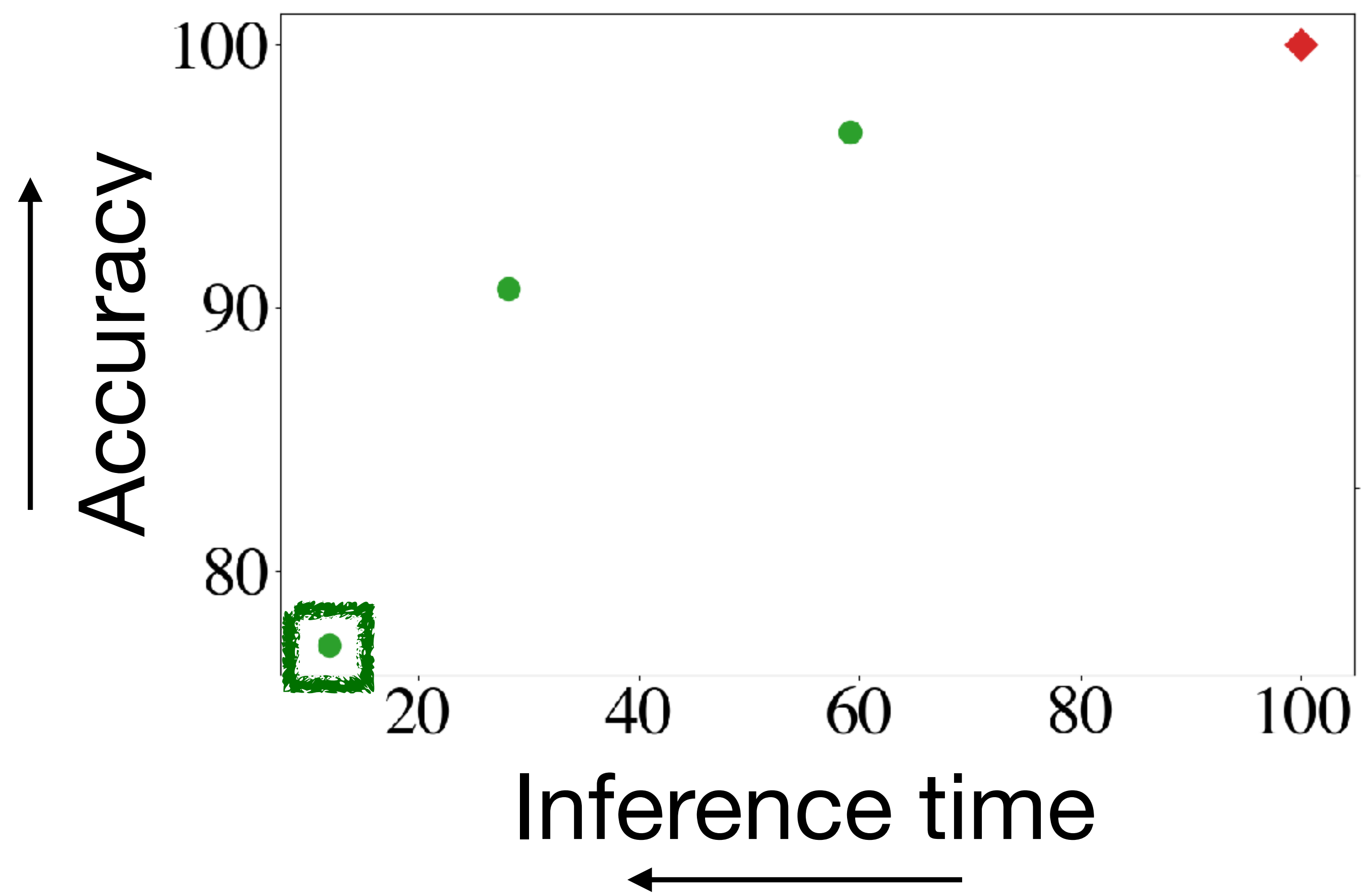
Schwartz et al., ACL 2020

*Some instances require **less processing** than others*

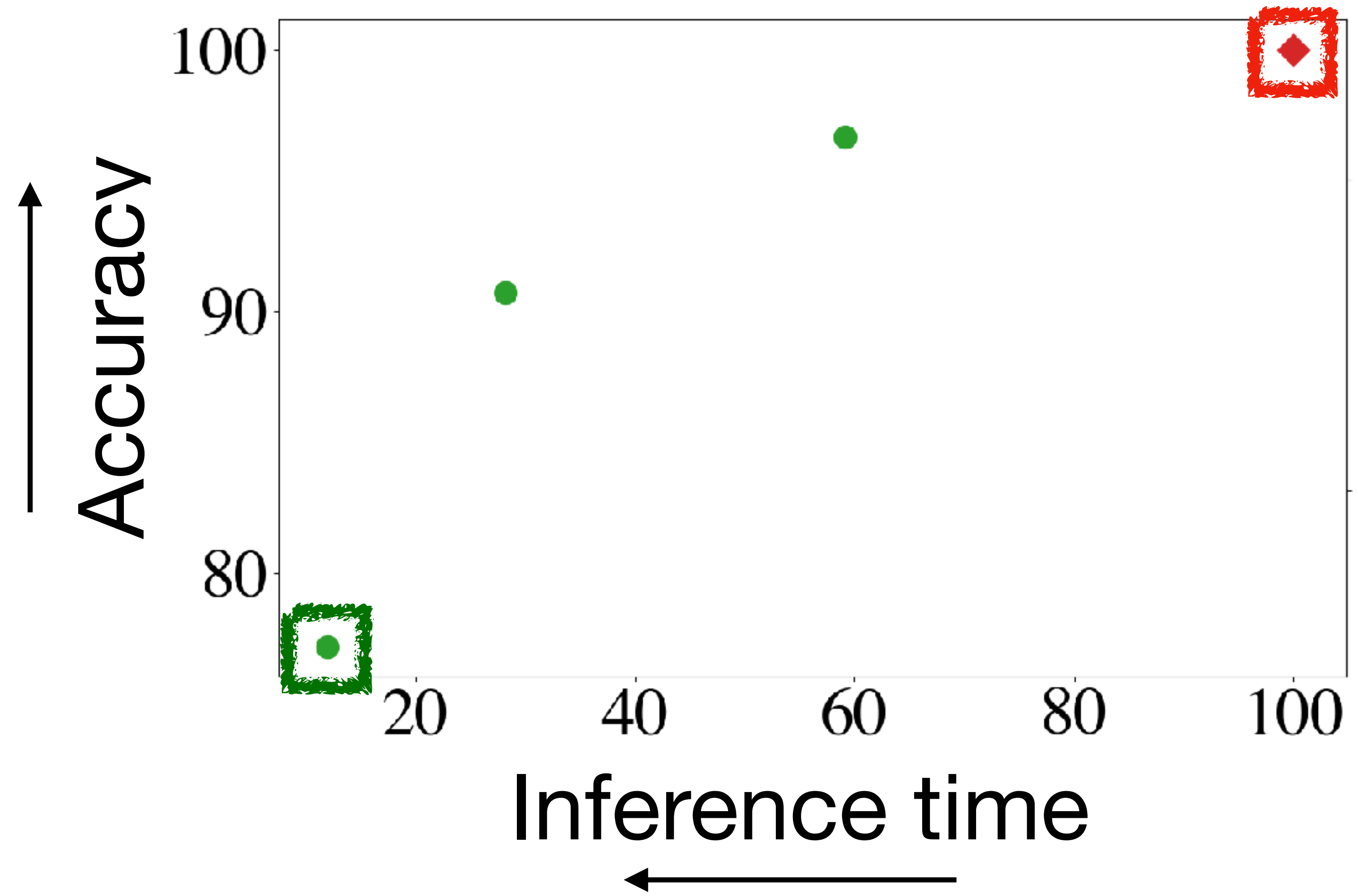
High-Level Idea



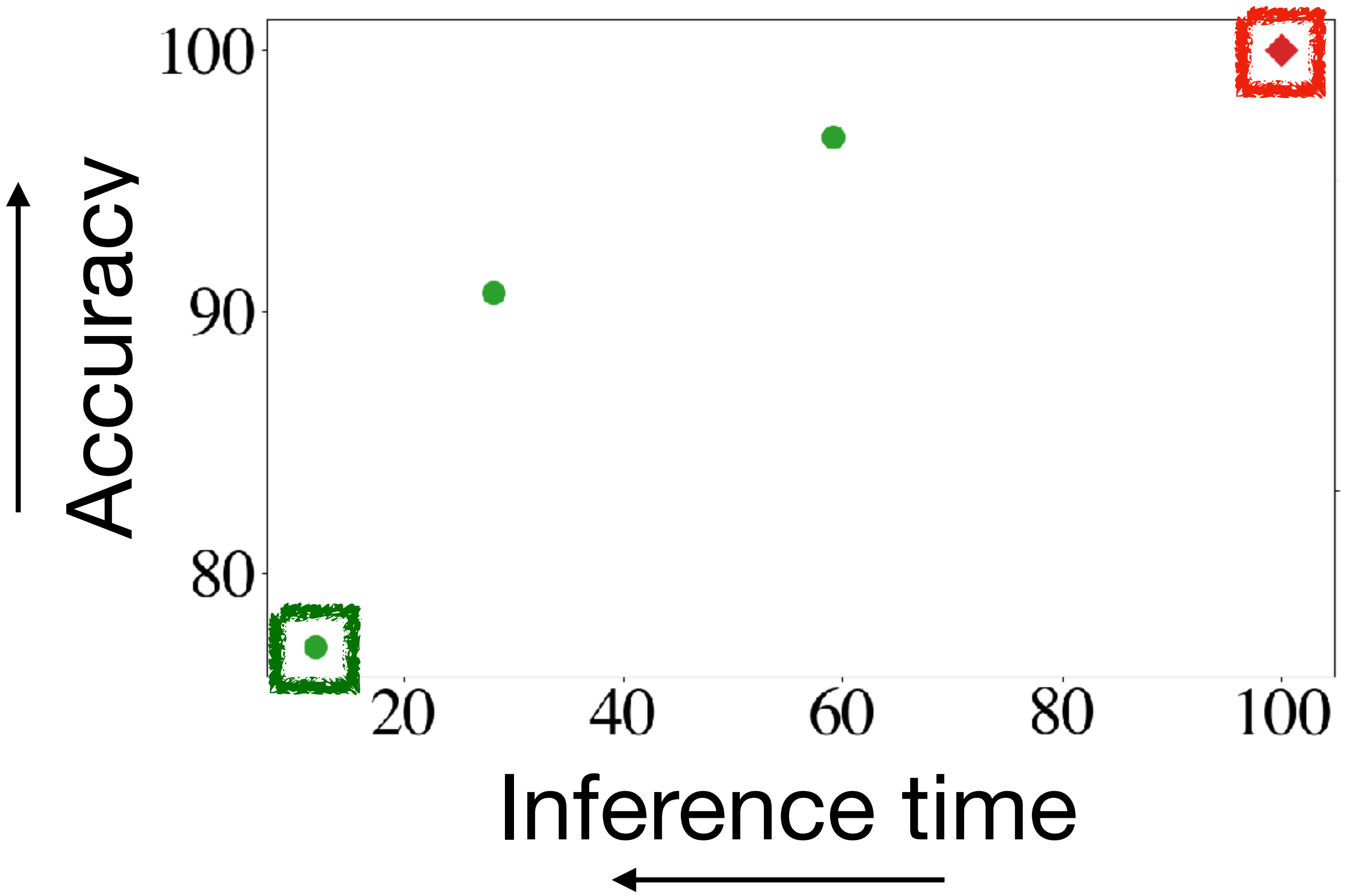
High-Level Idea



High-Level Idea



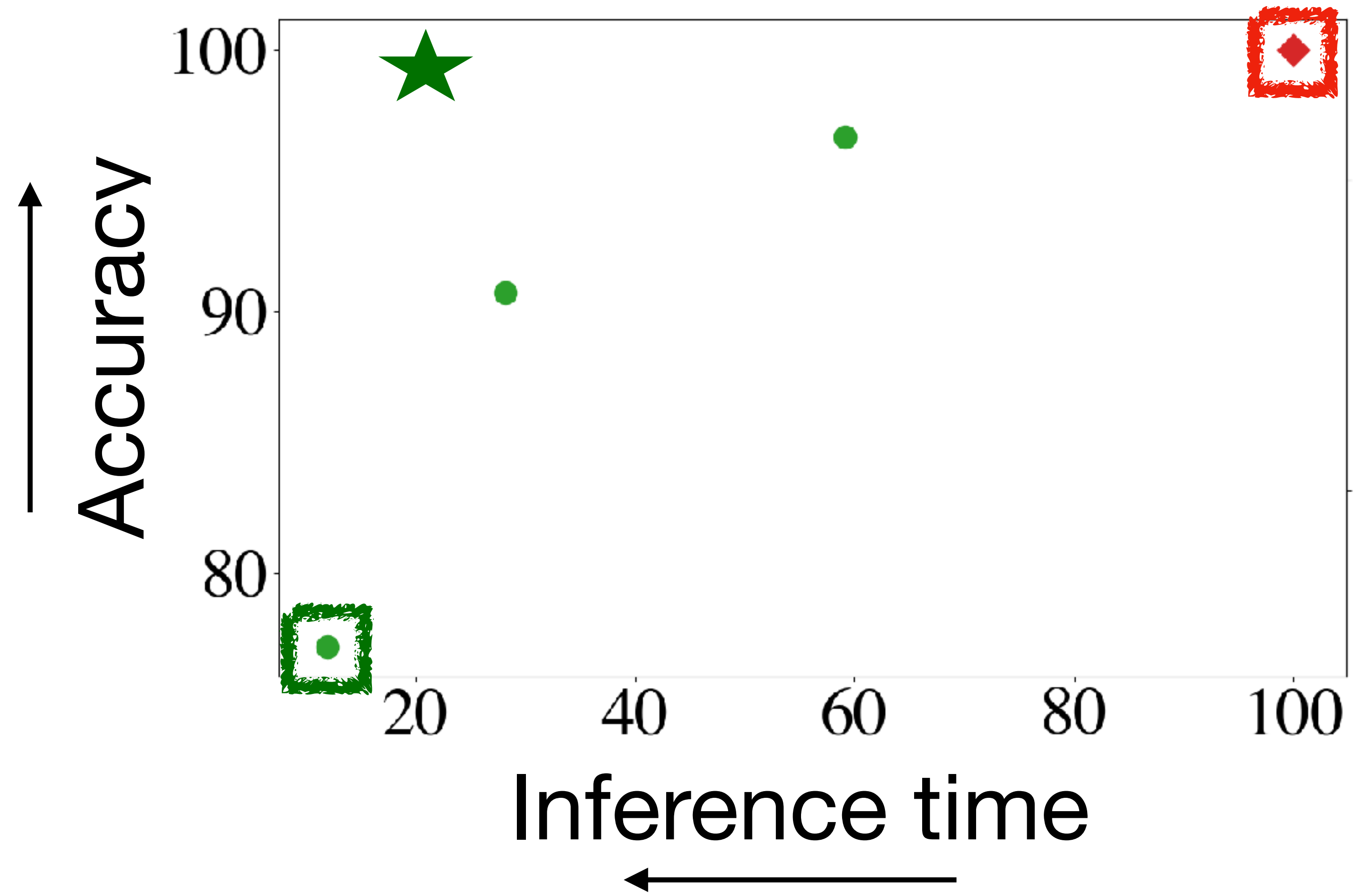
High-Level Idea



Run an efficient model on "easy" instances,

and an expensive model on "hard" instances

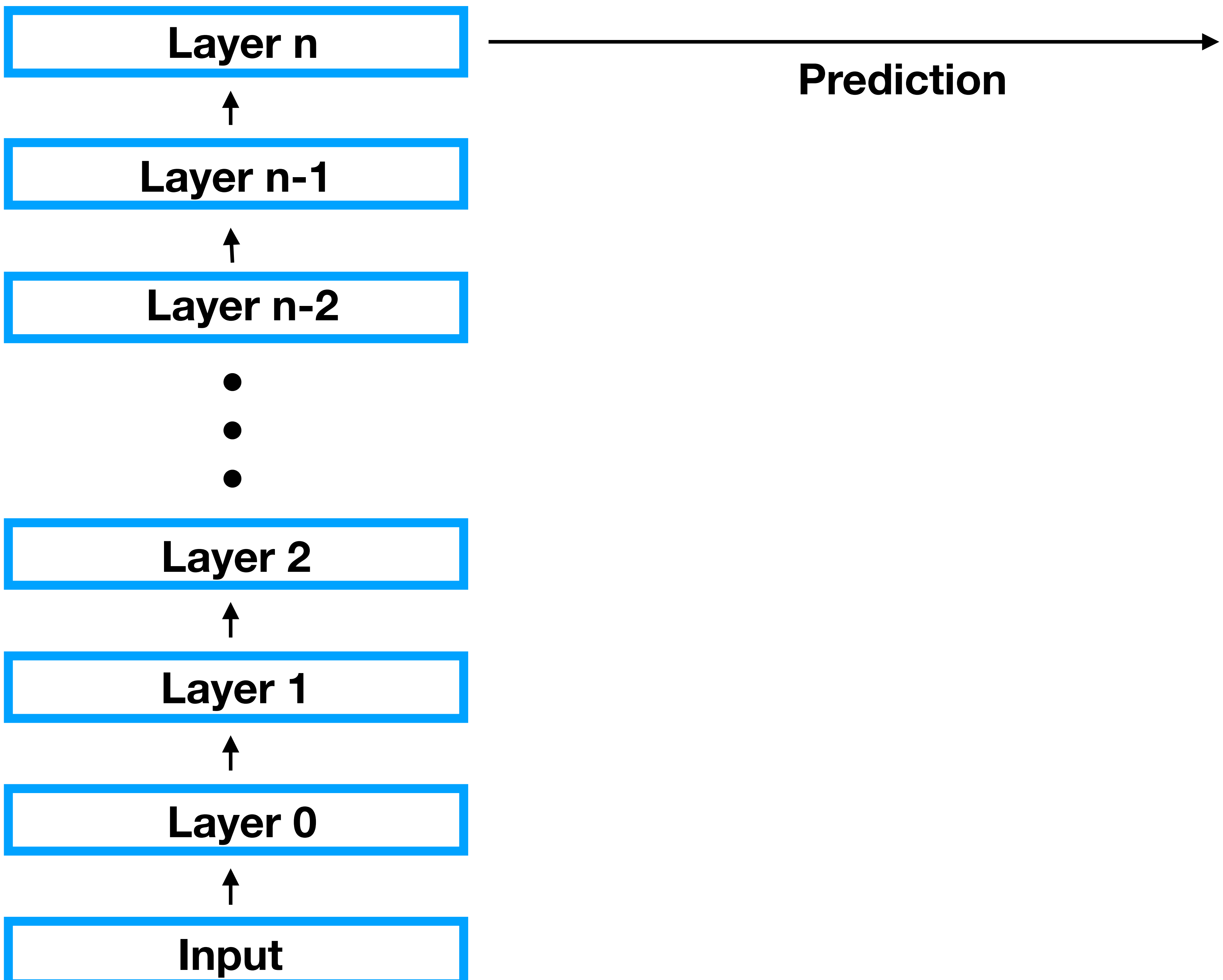
High-Level Idea



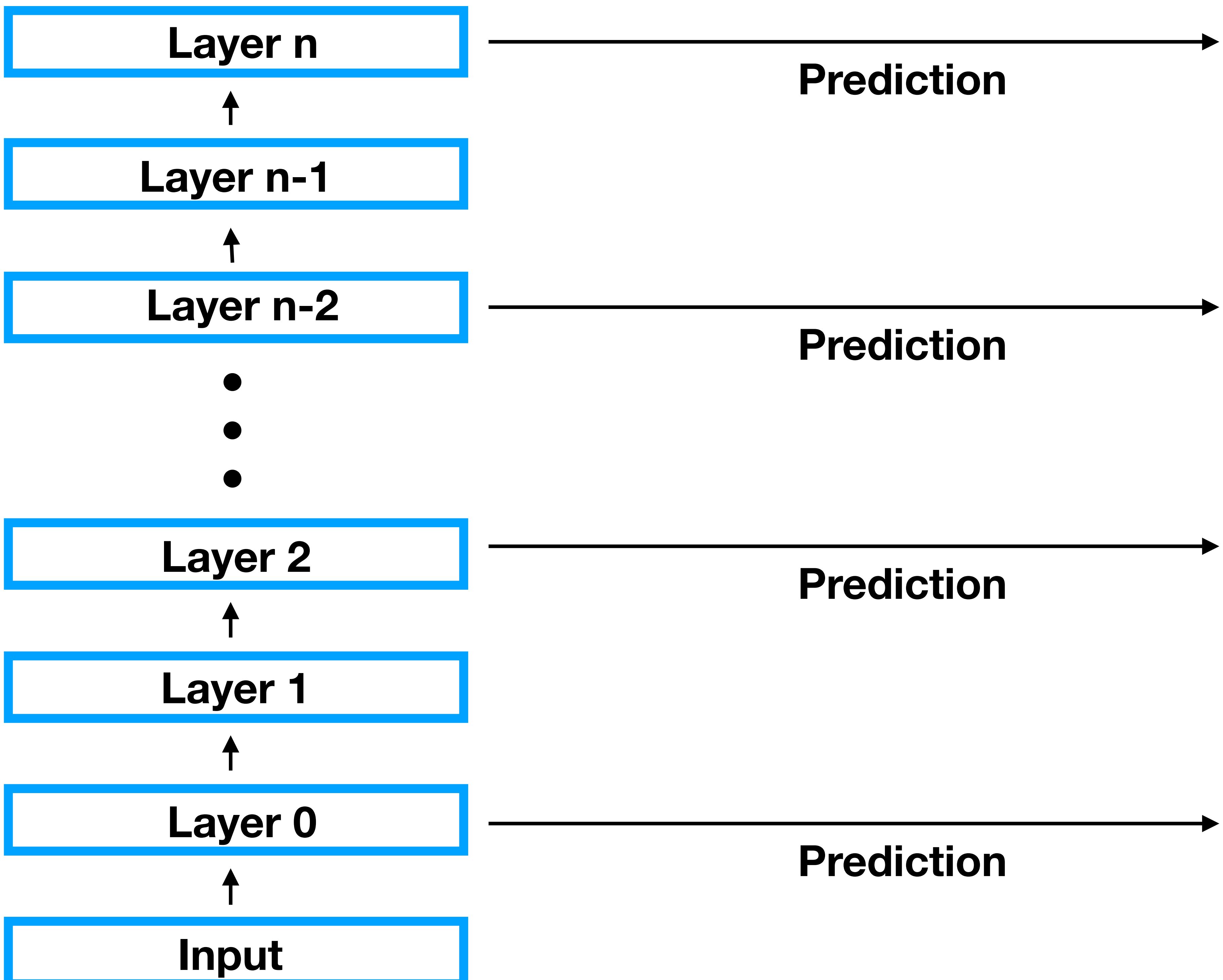
Run an efficient model on "easy" instances,

and an expensive model on "hard" instances

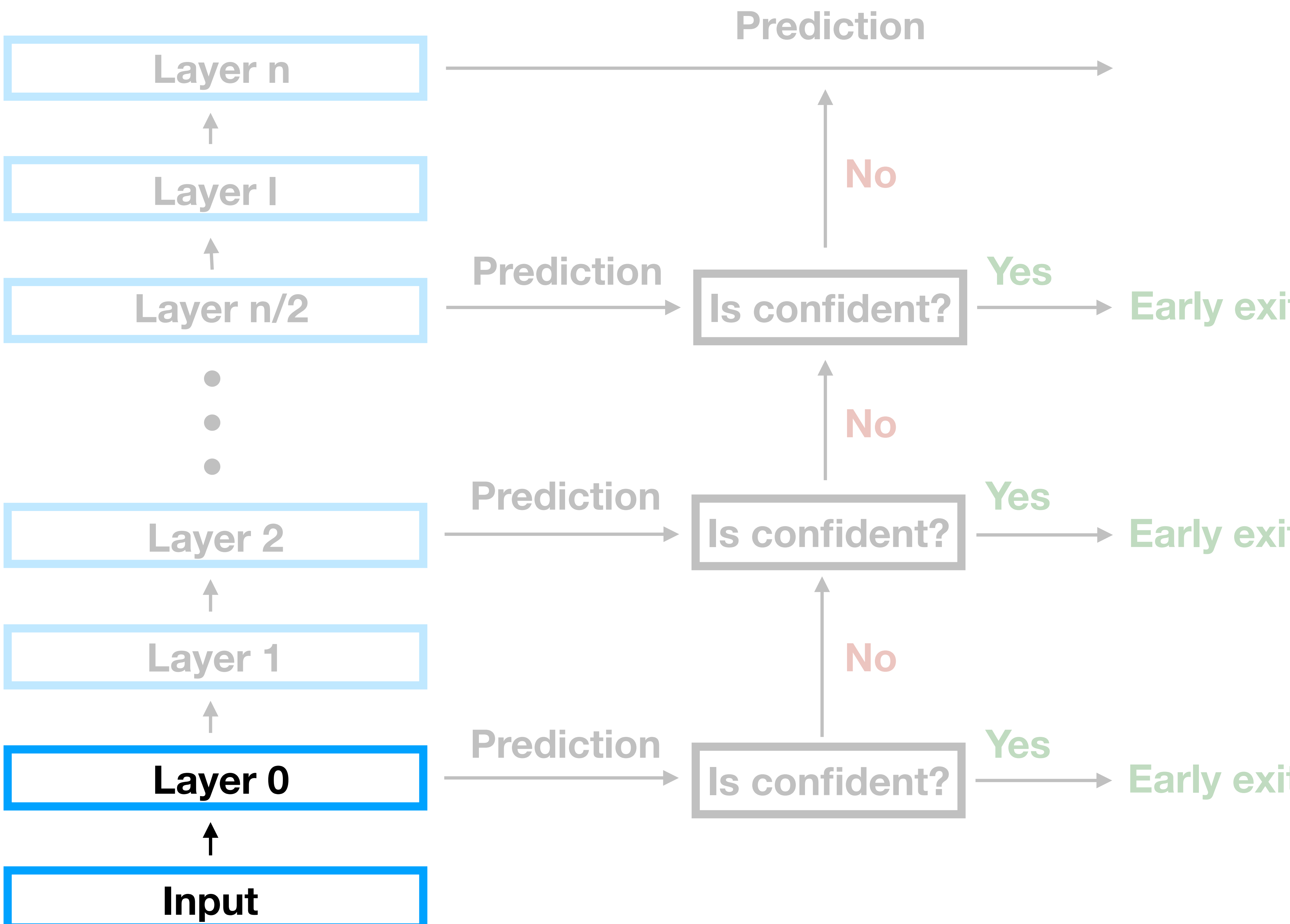
Our Approach: **Training Time**



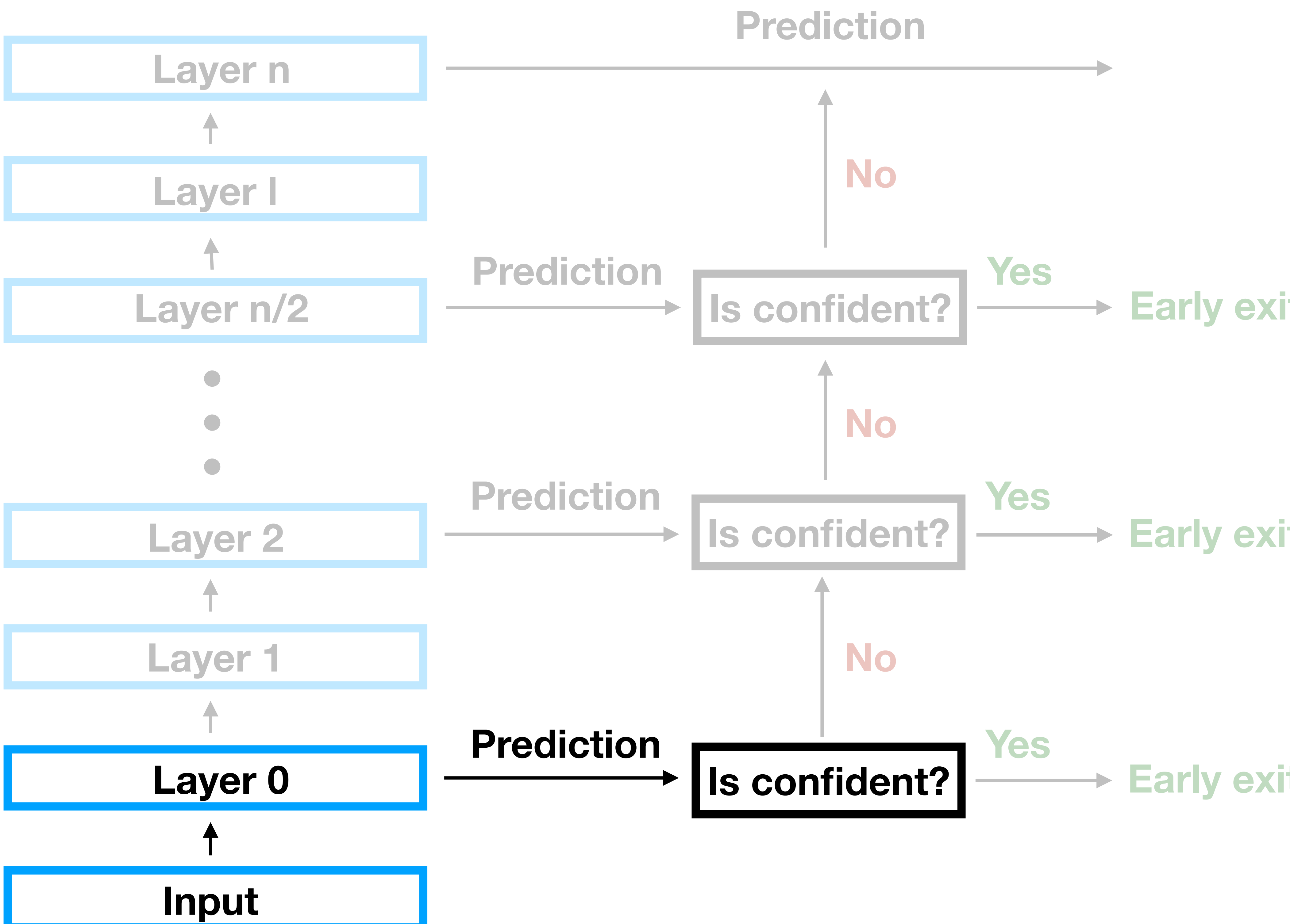
Our Approach: **Training Time**



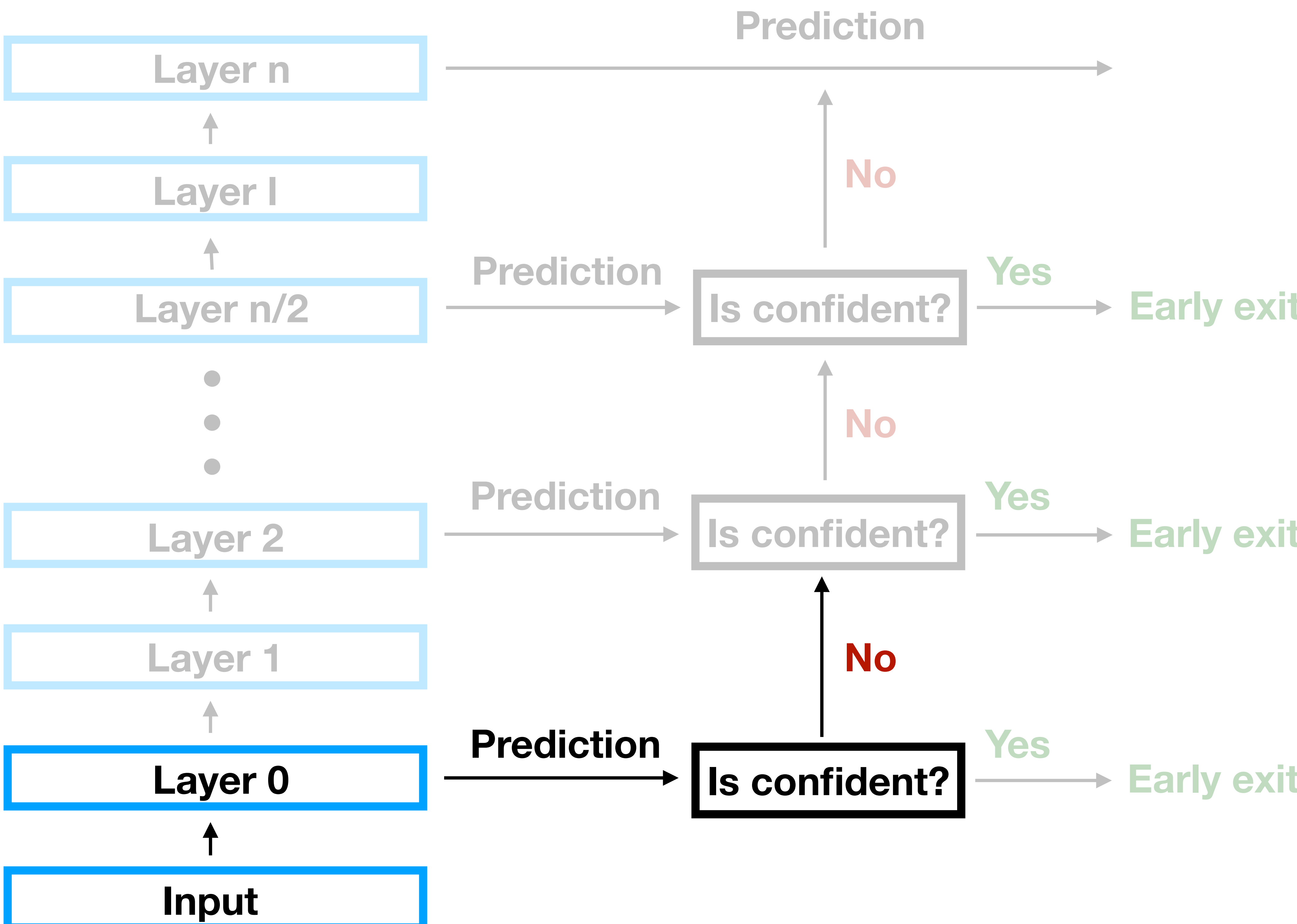
Our Approach: Test Time



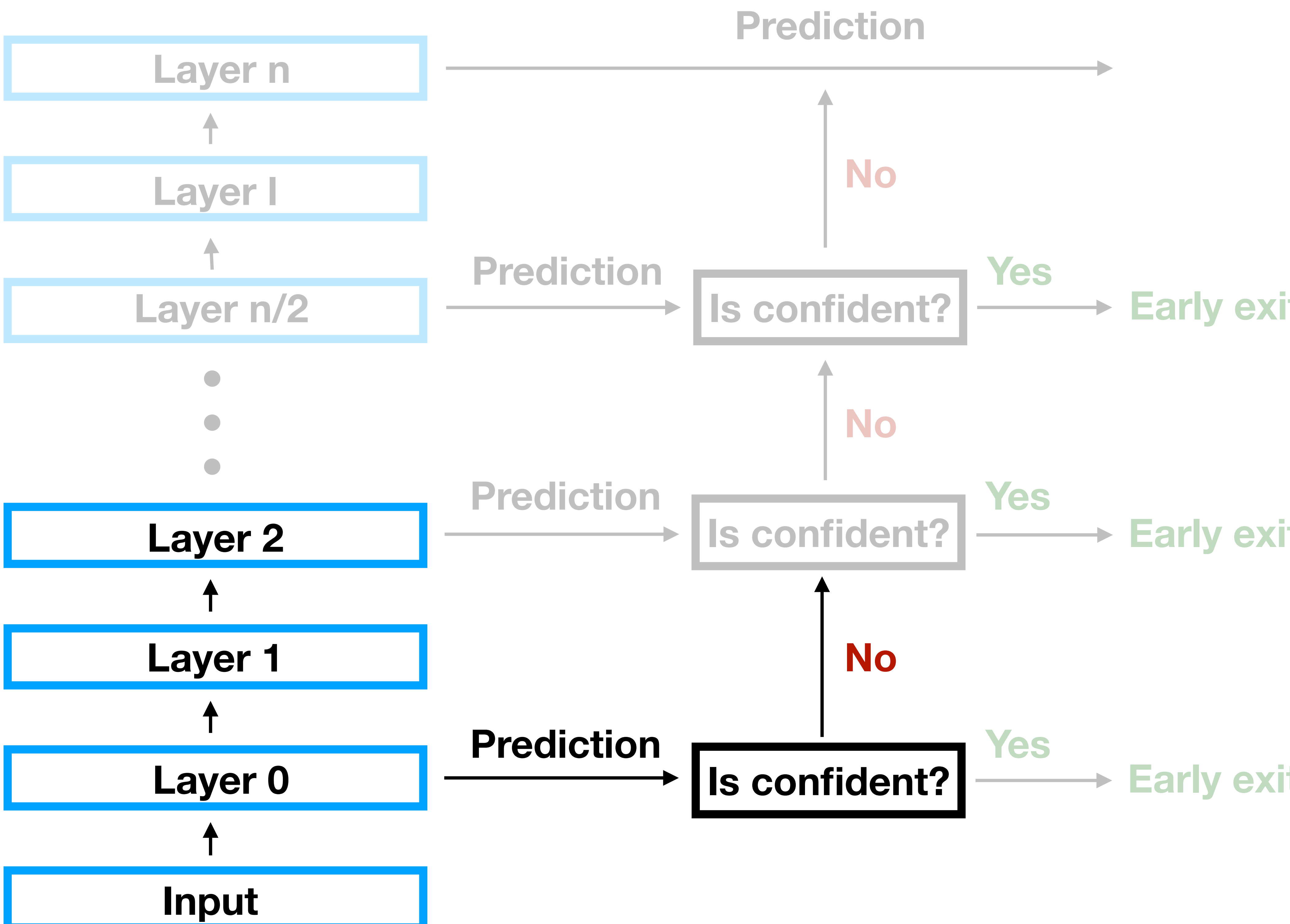
Our Approach: Test Time



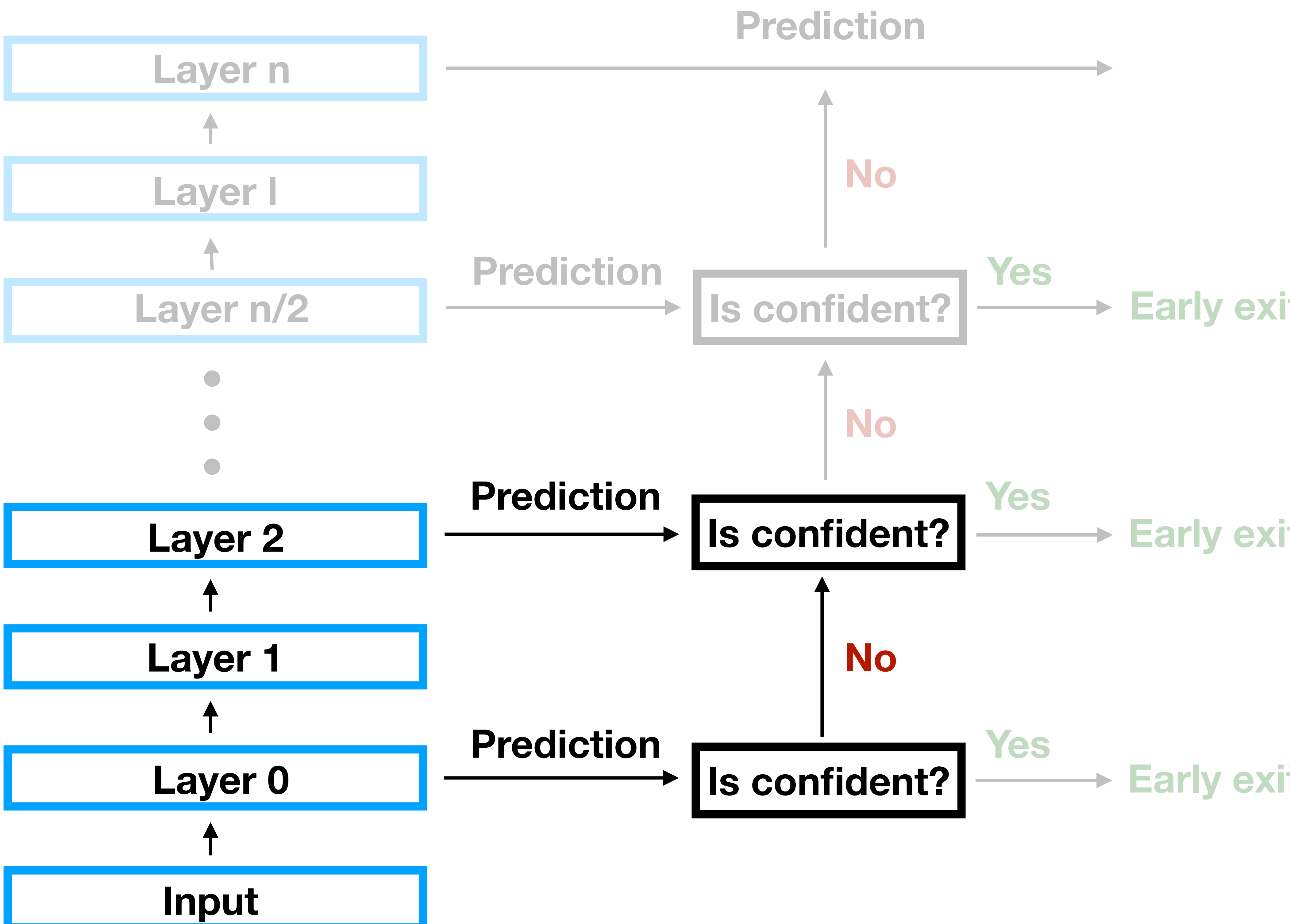
Our Approach: Test Time



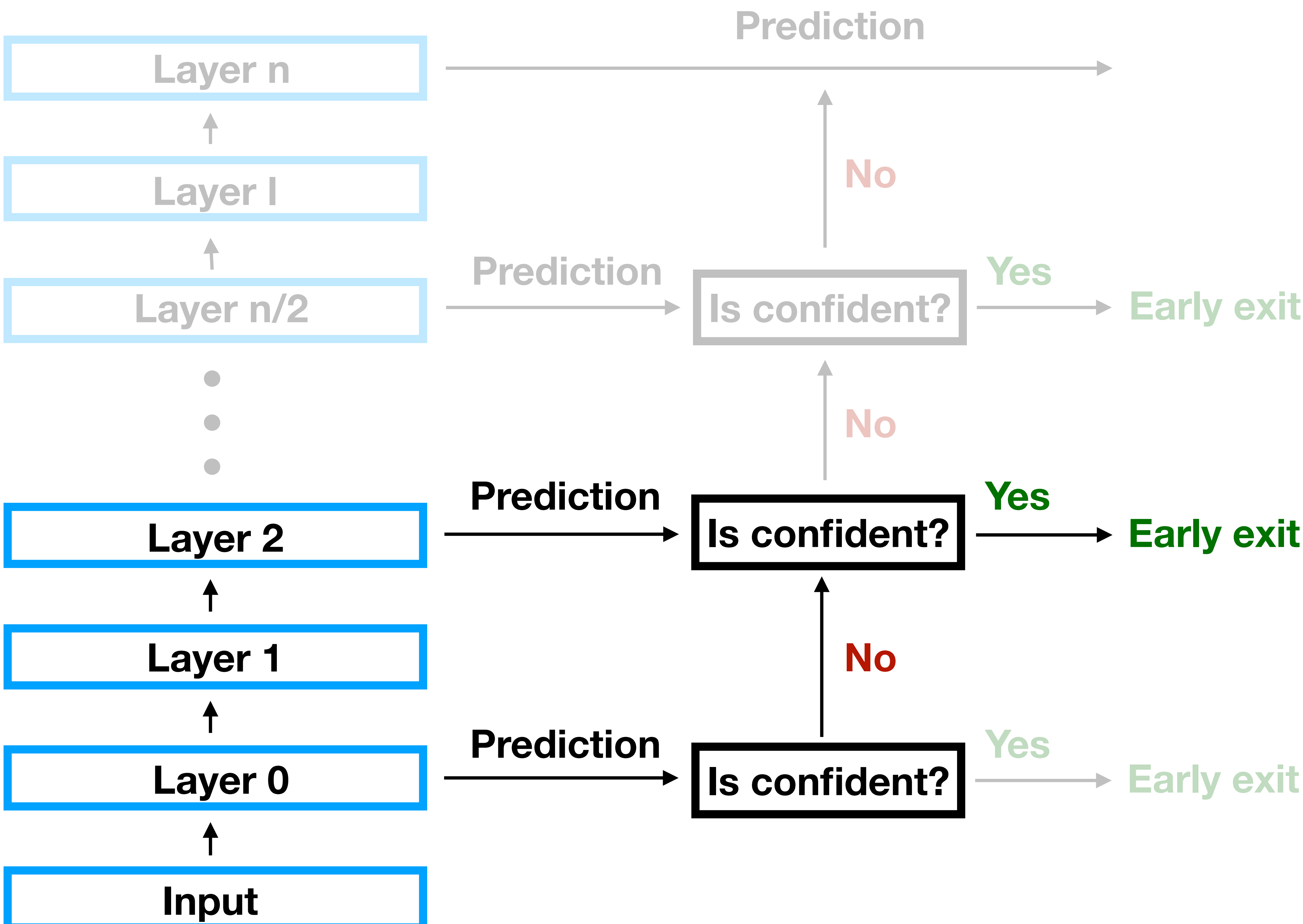
Our Approach: Test Time



Our Approach: Test Time



Our Approach: Test Time



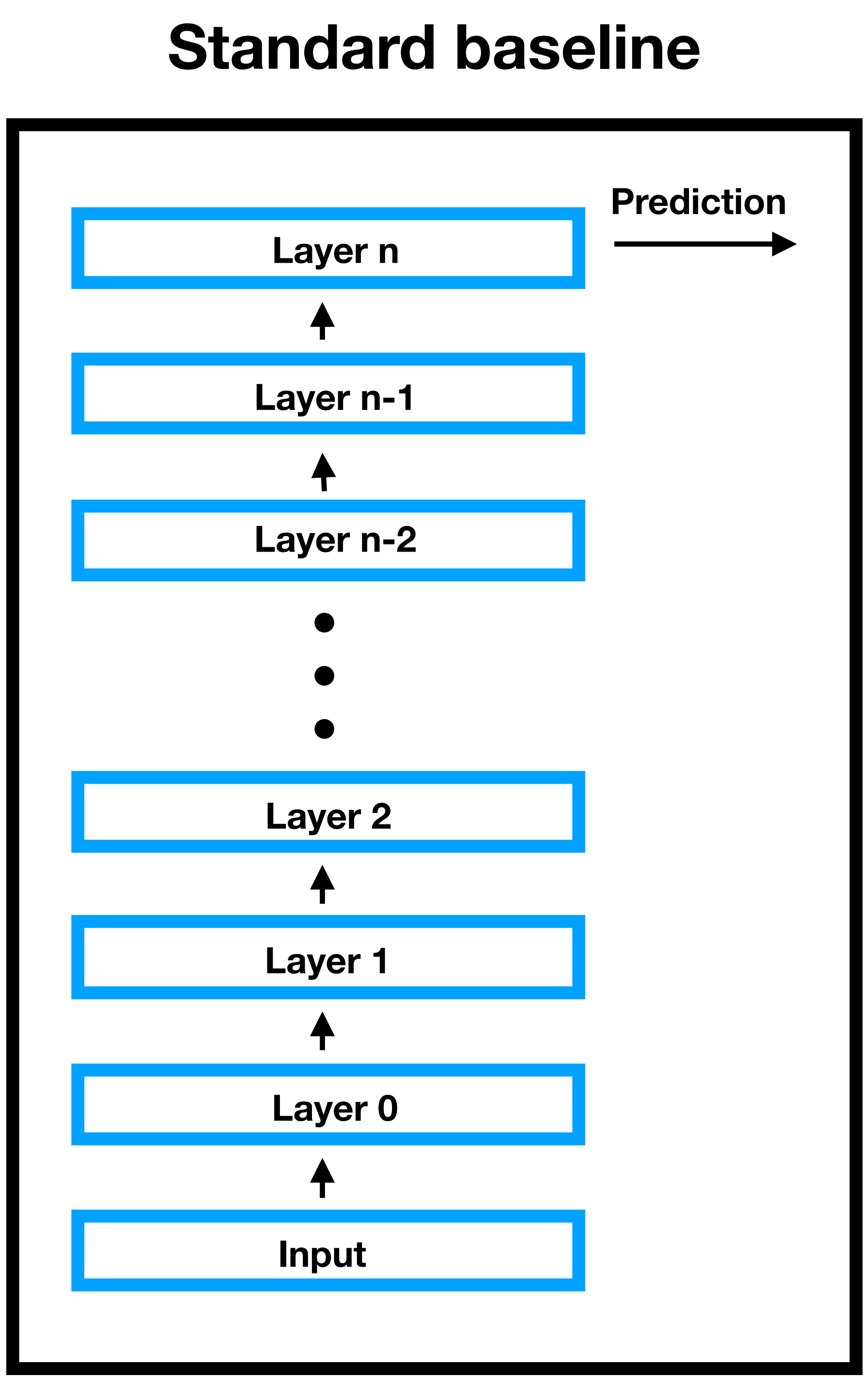
Calibrated Confidence Scores

- Interpret the calibrated softmax label scores as model confidence
  - We use temperature calibration (Guo et al., 2017)

Calibrated Confidence Scores

- Interpret the calibrated softmax label scores as model confidence
    - We use temperature calibration (Guo et al., 2017)
- Speed/accuracy tradeoff controlled by a **single early-exit confidence threshold**

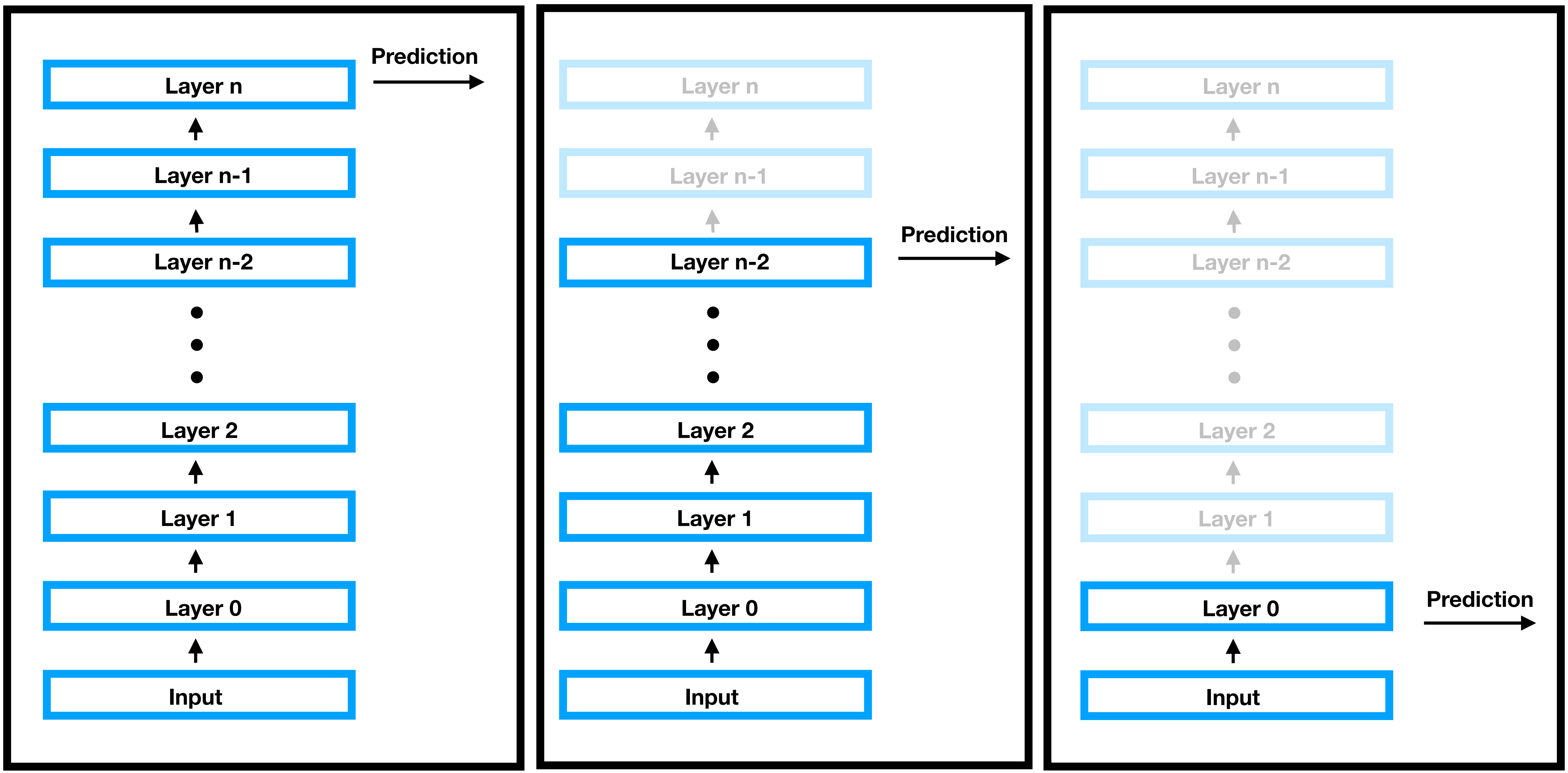
Baselines



Standard baseline
Prediction
Layer n
Layer n-1
Layer n-2
Layer 2
Layer 1
Layer 0
Input

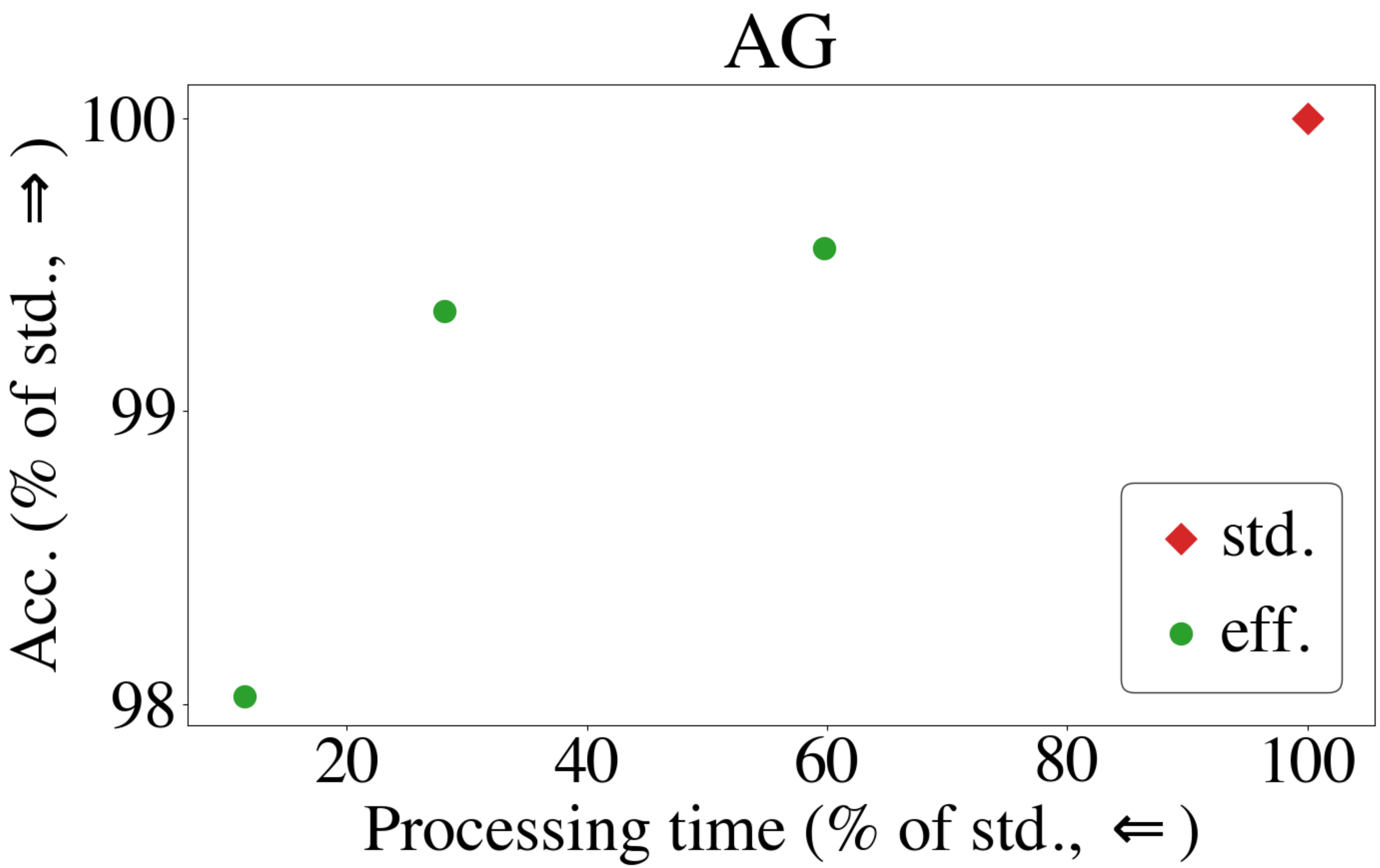
Baselines

Standard baseline **Efficient baselines**



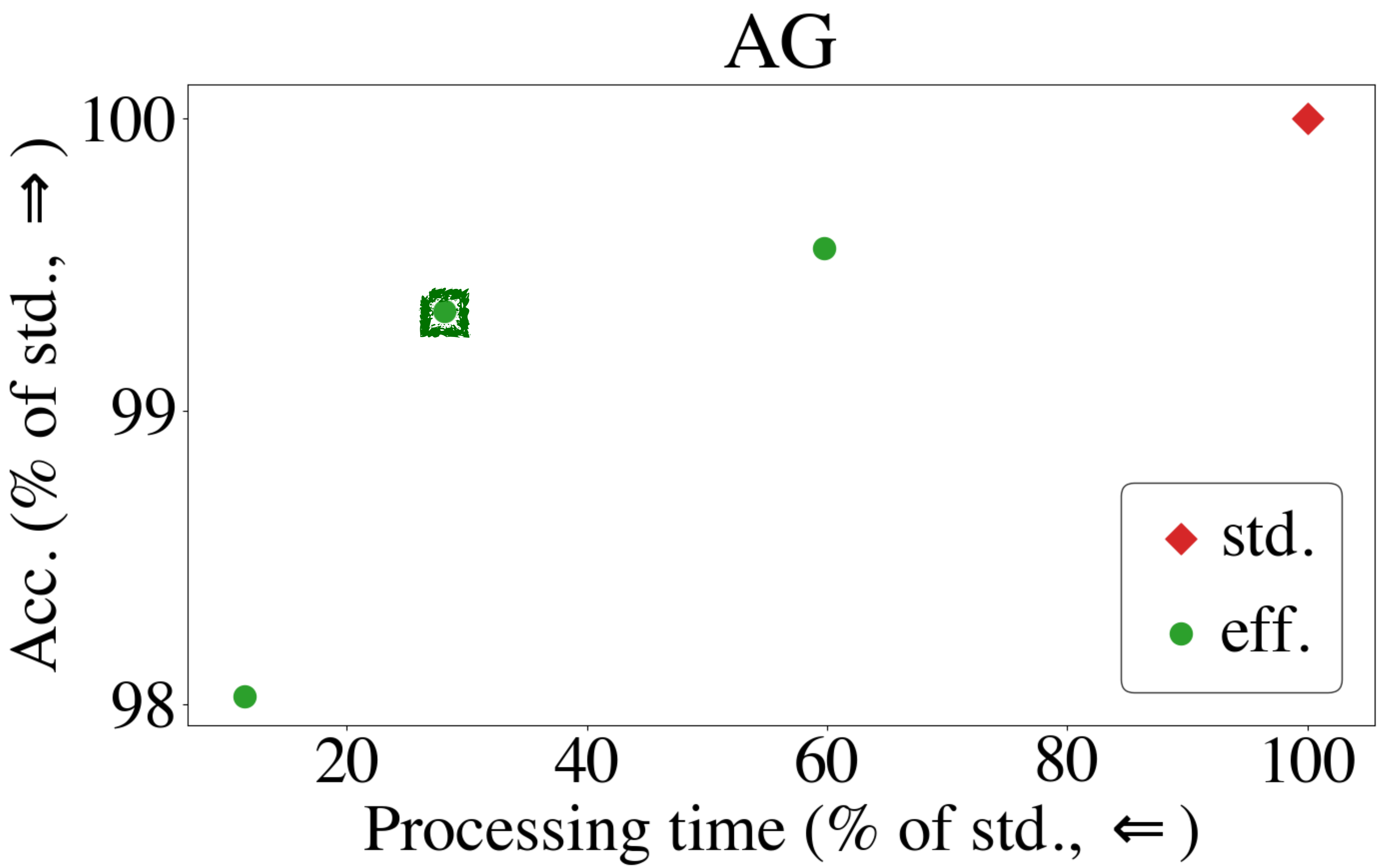
Experimental Results:

Strong Baselines!



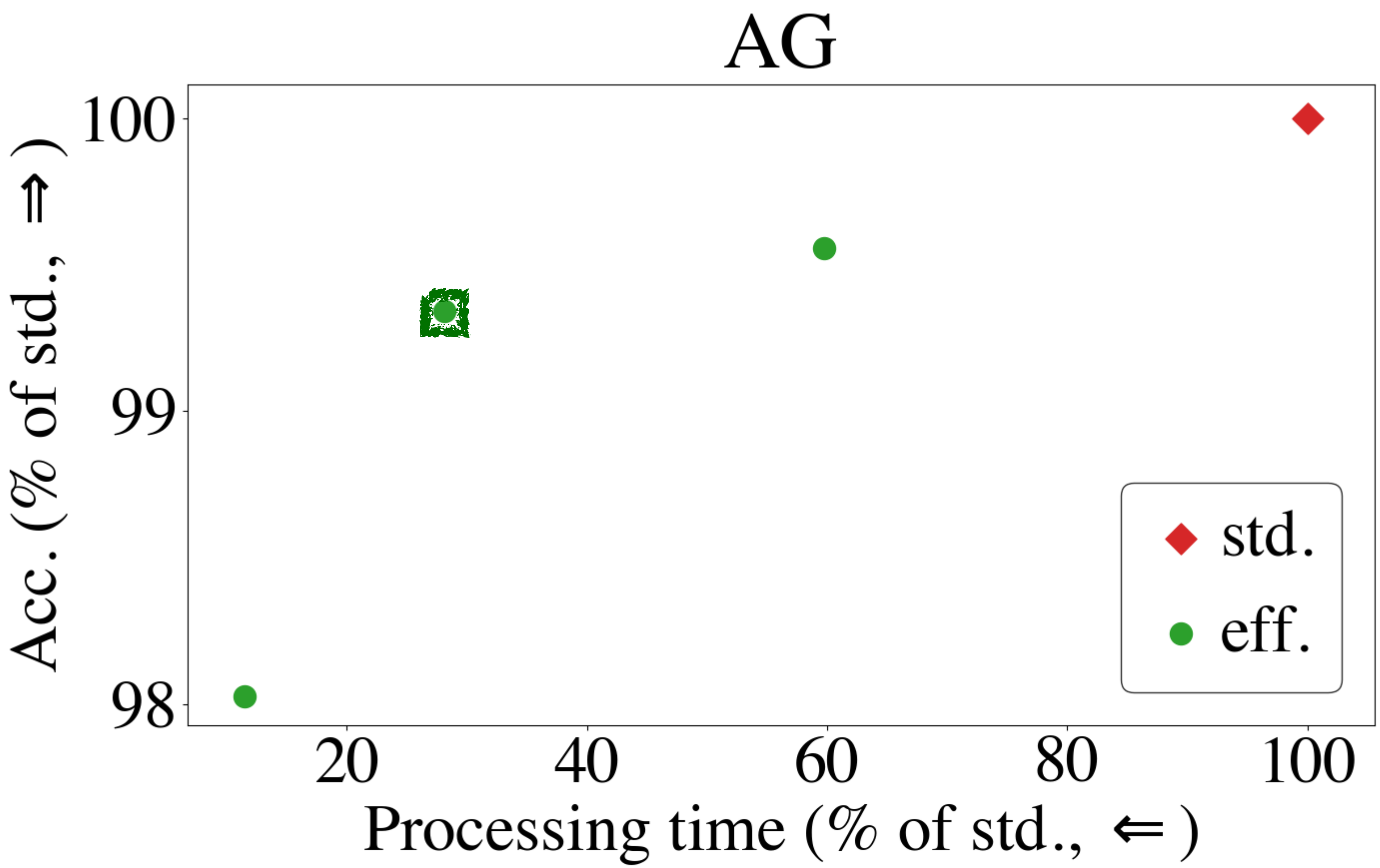
Experimental Results:

Strong Baselines!



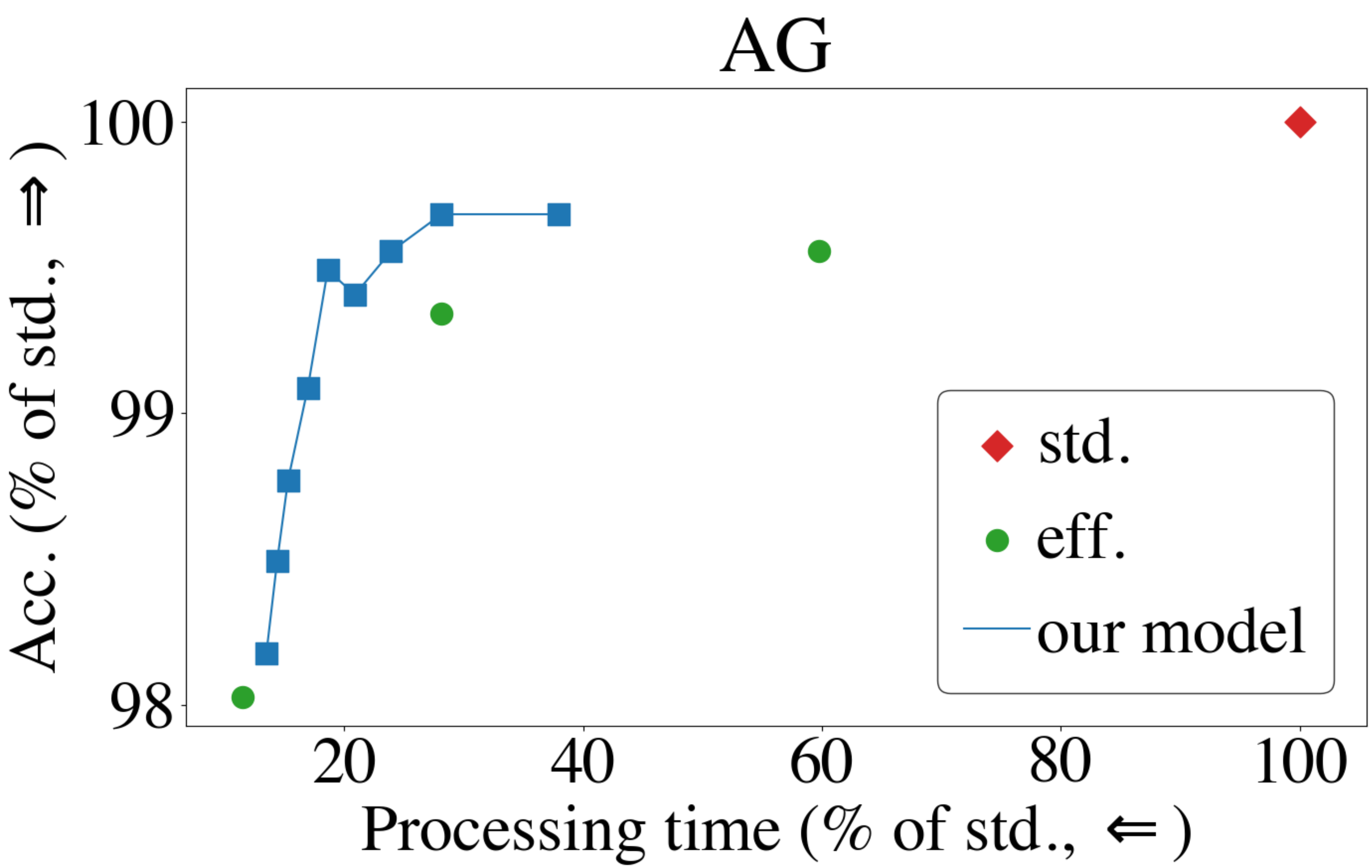
Experimental Results:

Strong Baselines!

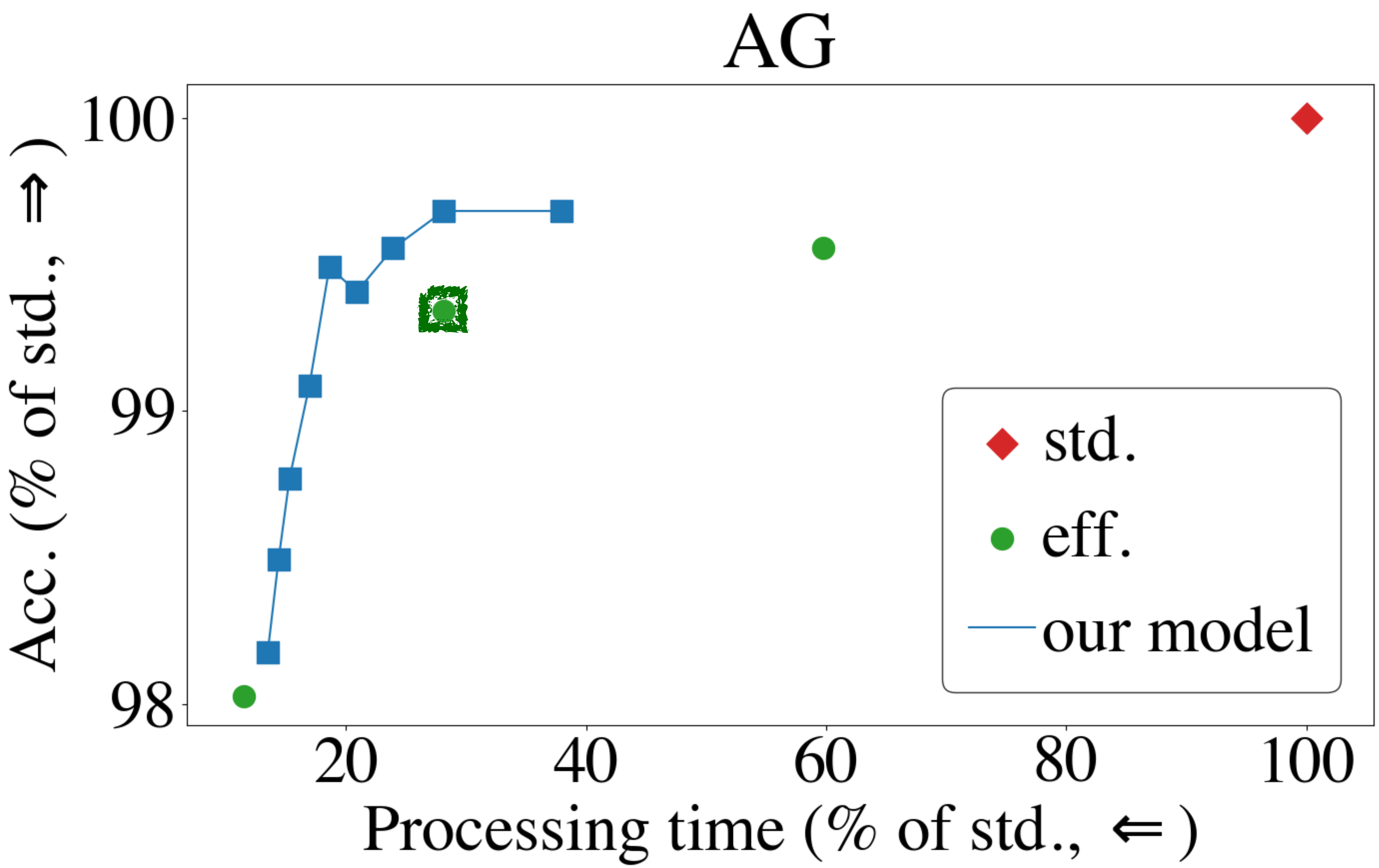


3 times faster, within 1% of full model

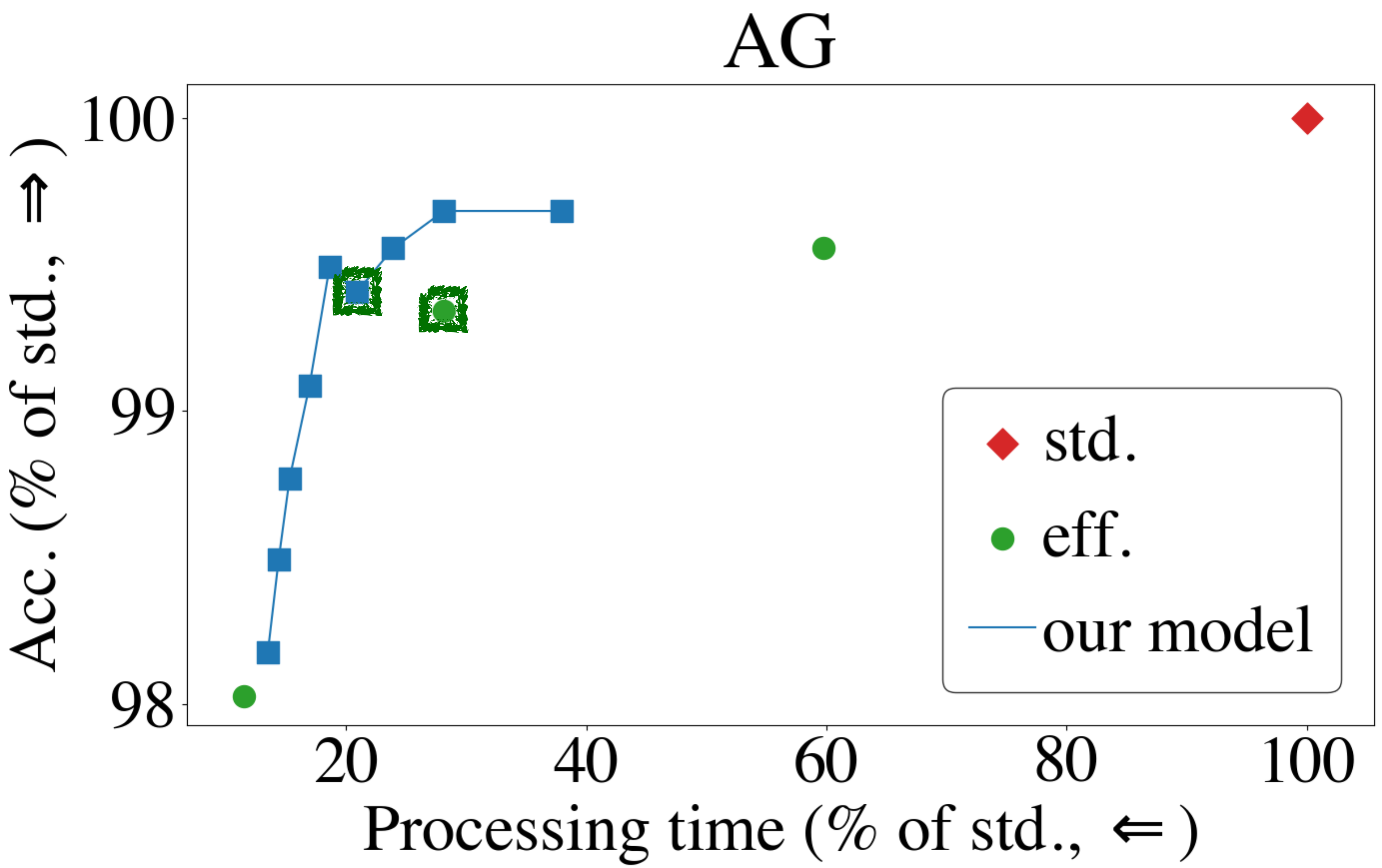
Better Speed/Accuracy Tradeoff



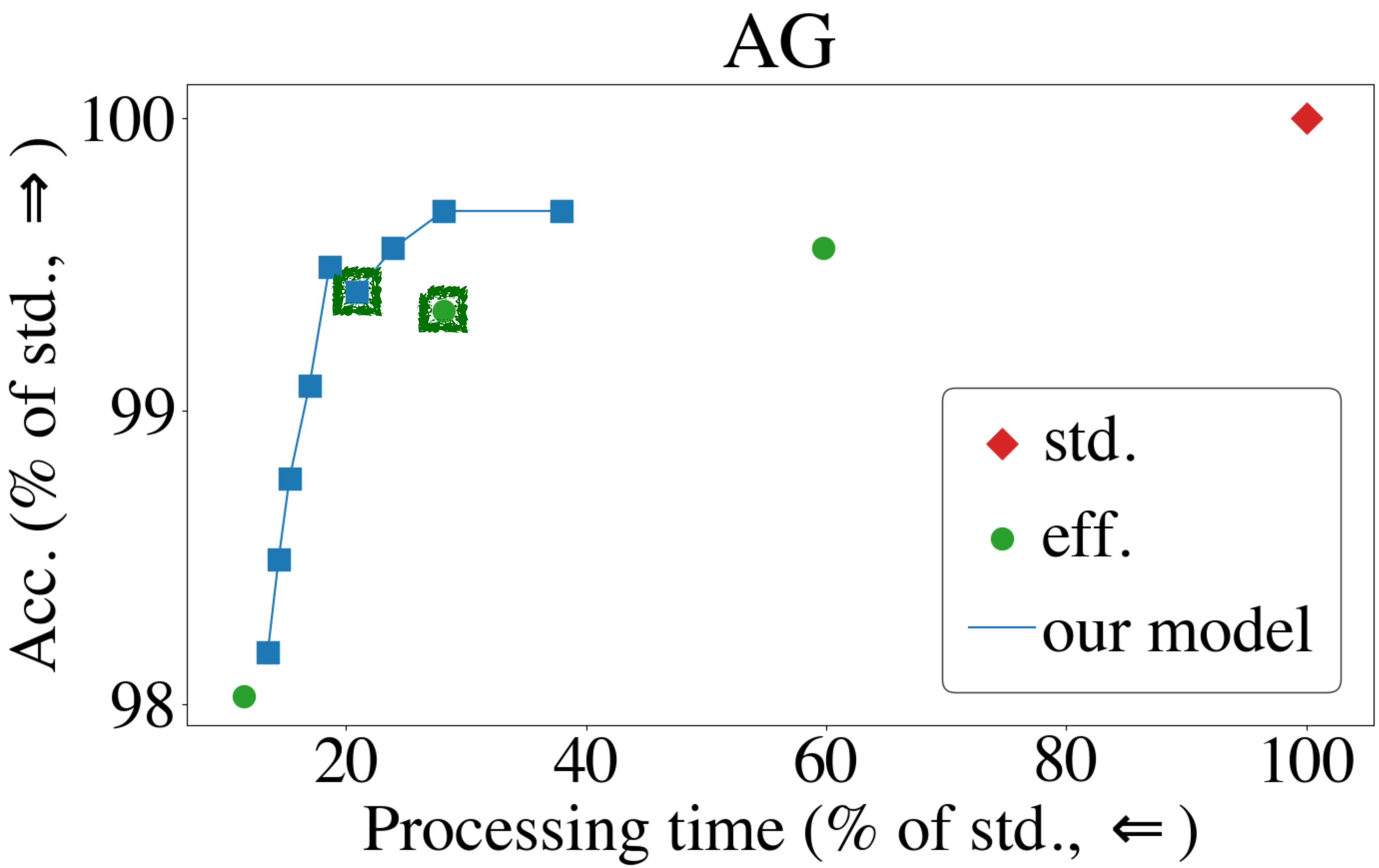
Better Speed/Accuracy Tradeoff



Better Speed/Accuracy Tradeoff

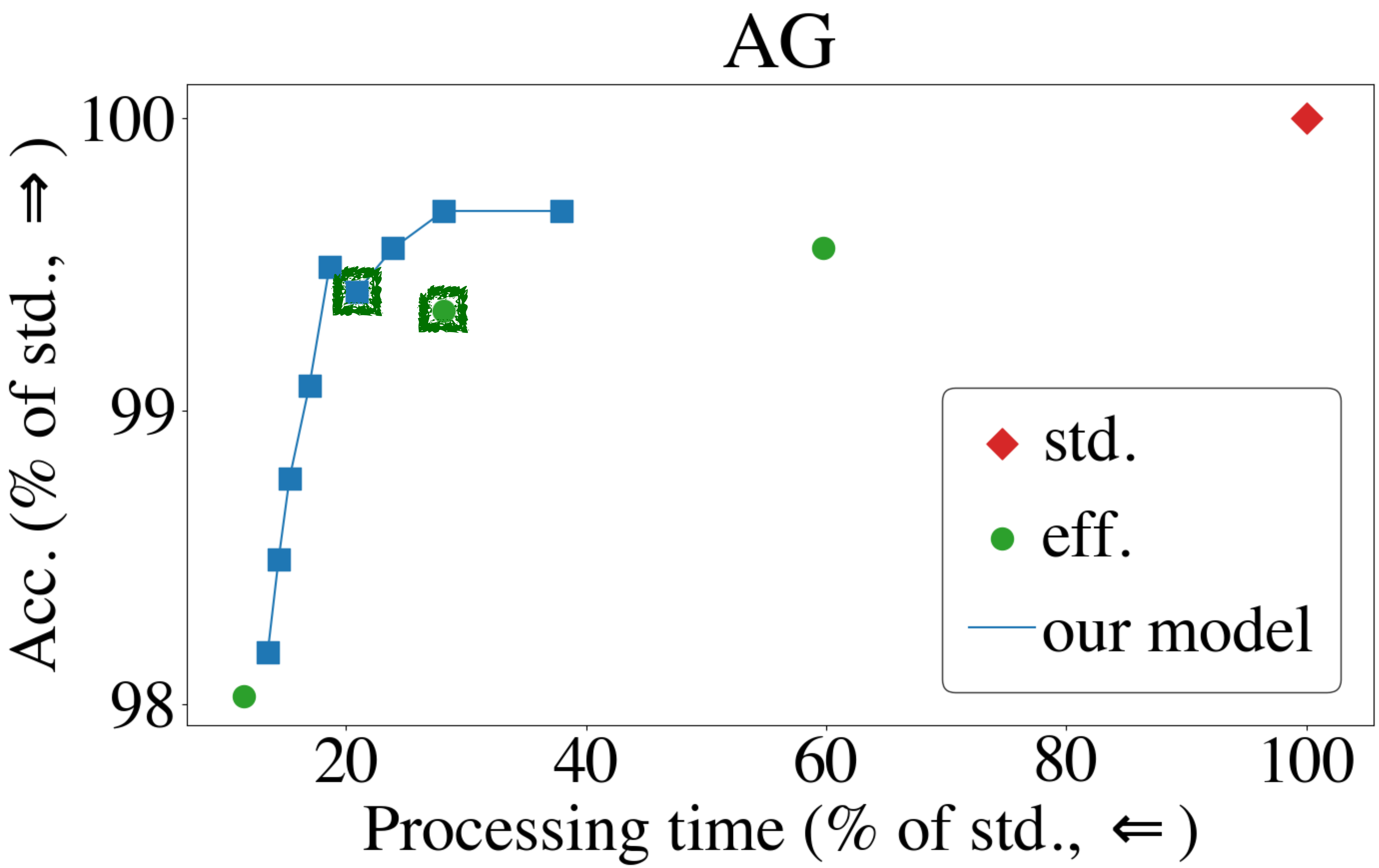


Better Speed/Accuracy Tradeoff



5 times faster, within 1% of full model

Better Speed/Accuracy Tradeoff



5 times faster, within 1% of full model

Similar results on SST, IMDB

More about our Approach

- No effective growth in parameters
  - $< 0.005\%$ additional parameters

More about our Approach

- No effective growth in parameters
    - < 0.005% additional parameters
- Training is **not** slower

More about our Approach

- No effective growth in parameters
  - < 0.005% additional parameters
- Training is **not** slower
- A **single** trained model provides multiple options along the speed/accuracy tradeoff
  - A single parameter: confidence threshold

More about our Approach

- No effective growth in parameters
  - < 0.005% additional parameters
- Training is **not** slower
- A **single** trained model provides multiple options along the speed/accuracy tradeoff
  - A single parameter: confidence threshold
- Caveat: requires batch size=1 during inference

Case Study 2: Efficient Training

Swayamdipta, **Schwartz** et al., EMNLP 2020

*Some instances are **more valuable** for training than others*

High-Level Idea

- *Divide the instances in a dataset into different groups*
- *Identify the groups that are **most valuable** for learning*
- *Train on those groups only, leading to **substantially faster training***

Training Dynamics

- Assume a model trained for K epochs

Training Dynamics

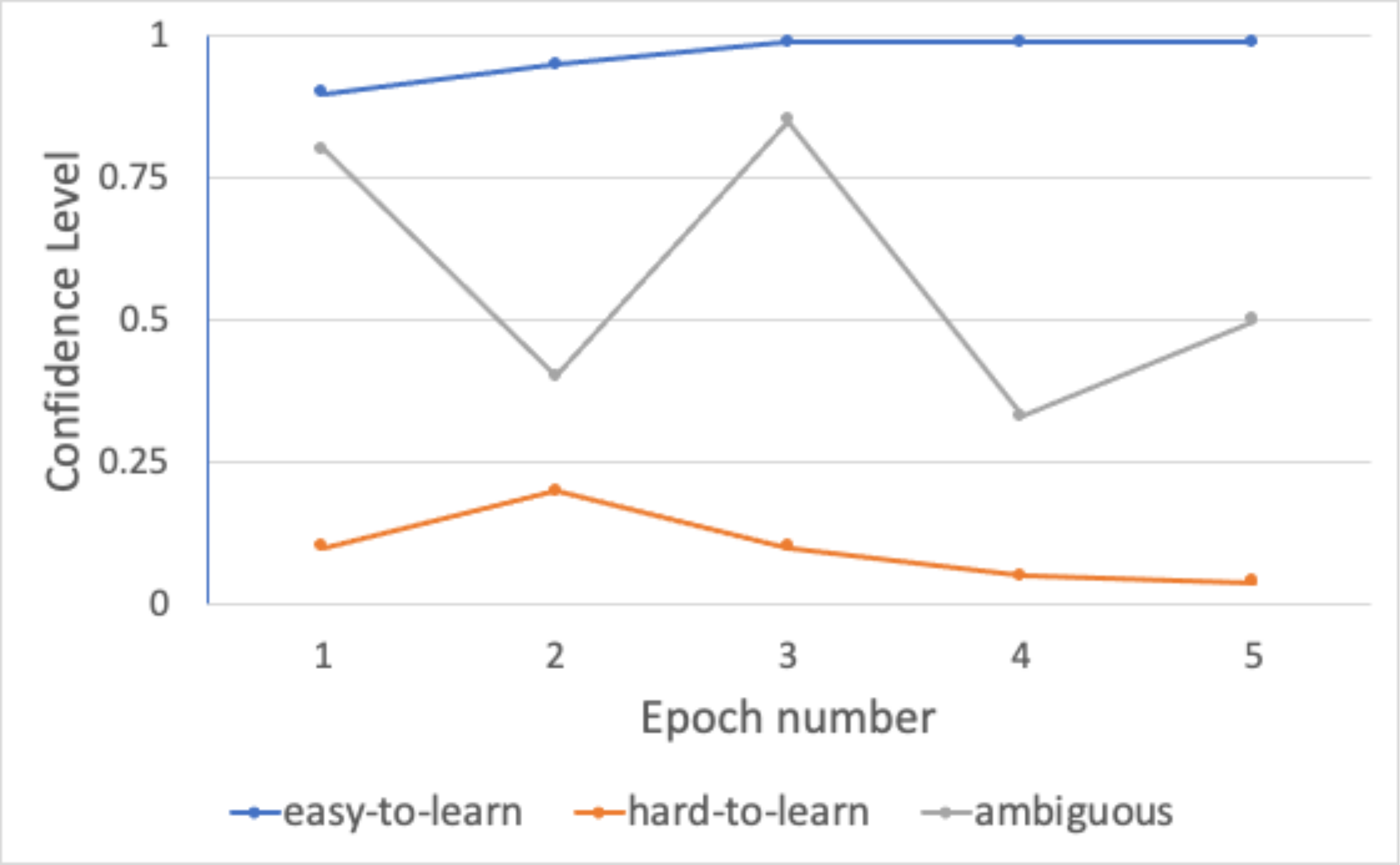
- Assume a model trained for K epochs
- At each epoch, the model makes predictions on each training sample
  - This leads to a vector of size K for each training instance

Training Dynamics

- Assume a model trained for K epochs
- At each epoch, the model makes predictions on each training sample
  - This leads to a vector of size K for each training instance
- We compute two measures on each vector:
  - Mean
  - Variability

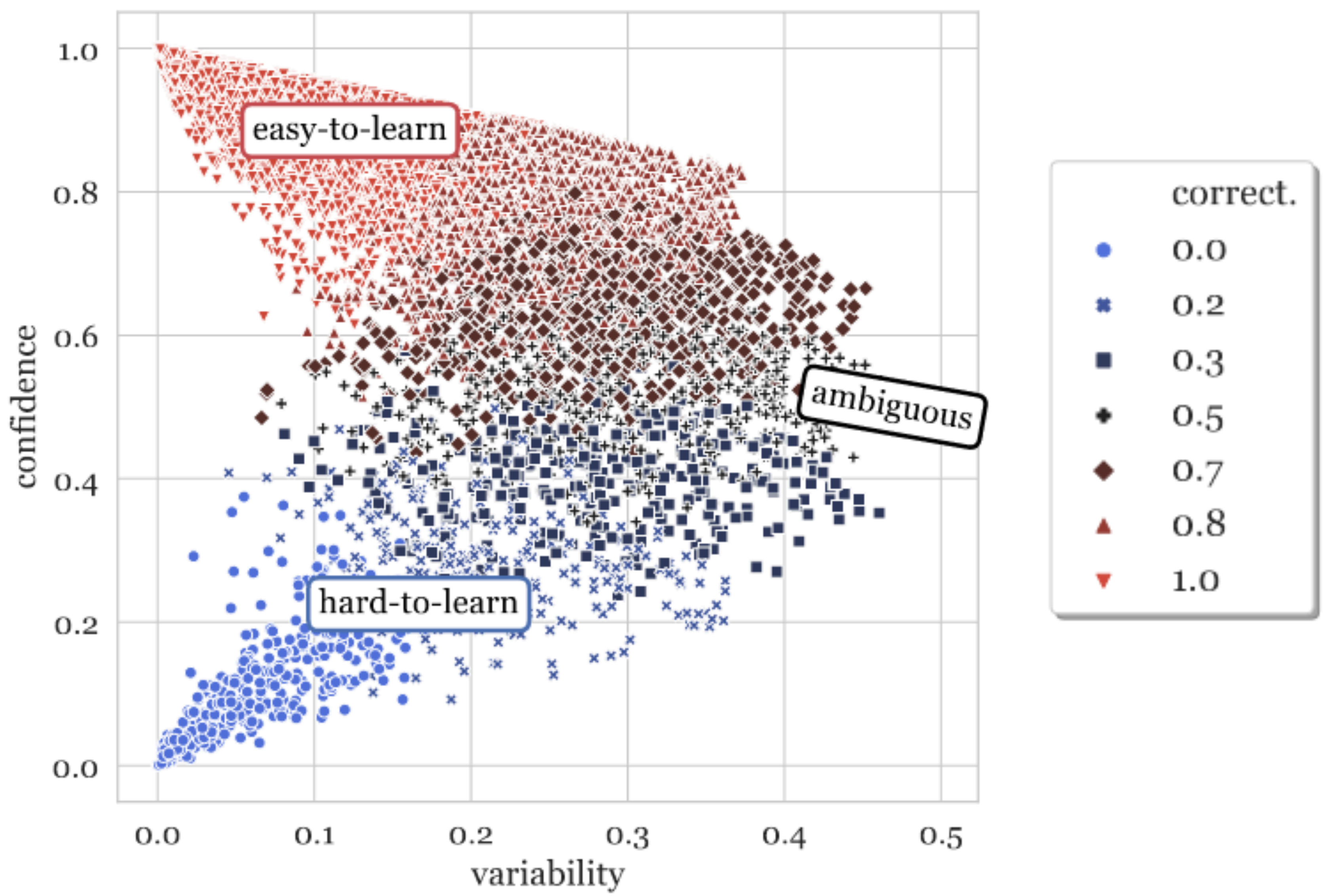
Training Dynamics

Toy Example



Dataset Map Example

SNLI, RoBERTa-Large



Sample-Efficient Training

- *easy-to-learn* instances provide little value to training
- Can we use training dynamics to select the *most valuable* instances?

Experiments

WinoGrande, RoBERTa-Large

	WinoG. Val. (ID)
100% train	$79.7_{0.2}$
random	$73.3_{1.3}$
forgetting	$75.5_{1.3}$
AL-uncertainty	$75.7_{0.8}$
AL-greedyK	$74.2_{0.4}$
AFLite	$76.8_{0.8}$
ambiguous	$\mathbf{78.7}_{0.4}$

Experiments

WinoGrande, RoBERTa-Large

	WinoG. Val. (ID)
100% train	$79.7_{0.2}$
random	$73.3_{1.3}$
forgetting	$75.5_{1.3}$
AL-uncertainty	$75.7_{0.8}$
AL-greedyK	$74.2_{0.4}$
AFLite	$76.8_{0.8}$
ambiguous	**78.7**$_{0.4}$

Experiments

WinoGrande, RoBERTa-Large

	WinoG. Val. (ID)
100% train	$79.7_{0.2}$
random	$73.3_{1.3}$
forgetting	$75.5_{1.3}$
AL-uncertainty	$75.7_{0.8}$
AL-greedyK	$74.2_{0.4}$
AFLite	$76.8_{0.8}$
ambiguous	$\mathbf{78.7}_{0.4}$

33%

Experiments

WinoGrande, RoBERTa-Large

	WinoG. Val. (ID)
100% train	$79.7_{0.2}$
random	$73.3_{1.3}$
forgetting	$75.5_{1.3}$
AL-uncertainty	$75.7_{0.8}$
AL-greedyK	$74.2_{0.4}$
AFLite	$76.8_{0.8}$
ambiguous	$\mathbf{78.7}_{0.4}$

33%

Experiments

WinoGrande, RoBERTa-Large

33%

	WinoG. Val. (ID)
100% train	$79.7_{0.2}$
random	$73.3_{1.3}$
forgetting	$75.5_{1.3}$
AL-uncertainty	$75.7_{0.8}$
AL-greedyK	$74.2_{0.4}$
AFLite	$76.8_{0.8}$
ambiguous	$\mathbf{78.7}_{0.4}$

Experiments

WinoGrande, RoBERTa-Large

		WinoG. Val. (ID)
	100% train	$79.7_{0.2}$
33%	random	$73.3_{1.3}$
	forgetting	$75.5_{1.3}$
	AL-uncertainty	$75.7_{0.8}$
	AL-greedyK	$74.2_{0.4}$
	AFLite	$76.8_{0.8}$
	ambiguous	$\mathbf{78.7}_{0.4}$

Experiments

WinoGrande, RoBERTa-Large

	WinoG. Val. (ID)	*WSC* (OOD)
100% train	$79.7_{0.2}$	$86.0_{0.1}$
random	$73.3_{1.3}$	$85.6_{0.4}$
forgetting	$75.5_{1.3}$	$84.8_{0.7}$
AL-uncertainty	$75.7_{0.8}$	$85.7_{0.8}$
AL-greedyK	$74.2_{0.4}$	$86.5_{0.5}$
AFLite	$76.8_{0.8}$	$86.6_{0.6}$
ambiguous	$\mathbf{78.7}_{0.4}$	$\mathbf{87.6}_{0.6}$

33%

Experiments

WinoGrande, RoBERTa-Large

	WinoG. Val. (ID)	*WSC* (OOD)
100% train	$79.7_{0.2}$	$86.0_{0.1}$
random	$73.3_{1.3}$	$85.6_{0.4}$
forgetting	$75.5_{1.3}$	$84.8_{0.7}$
AL-uncertainty	$75.7_{0.8}$	$85.7_{0.8}$
AL-greedyK	$74.2_{0.4}$	$86.5_{0.5}$
AFLite	$76.8_{0.8}$	$86.6_{0.6}$
ambiguous	$\mathbf{78.7}_{0.4}$	$\mathbf{87.6}_{0.6}$

33%

Experiments

WinoGrande, RoBERTa-Large

	WinoG. Val. (ID)	*WSC* (OOD)
100% train	$79.7_{0.2}$	$86.0_{0.1}$
random	$73.3_{1.3}$	$85.6_{0.4}$
forgetting	$75.5_{1.3}$	$84.8_{0.7}$
AL-uncertainty	$75.7_{0.8}$	$85.7_{0.8}$
AL-greedyK	$74.2_{0.4}$	$86.5_{0.5}$
AFLite	$76.8_{0.8}$	$86.6_{0.6}$
ambiguous	$\mathbf{78.7}_{0.4}$	$\mathbf{87.6}_{0.6}$

33%

Similar results on *SNLI, MNLI, QNLI*

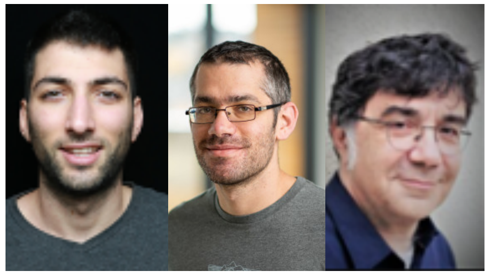
Recap

- Some instances contribute more to learning
- We select the ones with the *highest variance* in confidence level across training
- 3x reduction in training time
  - Minimal reduction in ID performance
  - **Improvement** on OOD performance
- Limitations
  - Model-dependent
  - Requires training on the full dataset first

Case Study 3:
Efficient Pre-training for Vision and Language

Bitton, Stanovsky, Elhadad & **Schwartz**, Findings of EMNLP 2021

Some ***words*** *are* ***more valuable*** *for pre-training than others*



MLM in Vision and Language

- Virtually all vision pre-training works (Shin et al., 2021) follow BERT and randomly mask 15% of the tokens

MLM in Vision and Language

- Virtually all vision pre-training works (Shin et al., 2021) follow BERT and randomly mask 15% of the tokens
- Of the masked tokens, roughly one half are *stop-words* or *punctuation*

MLM in Vision and Language

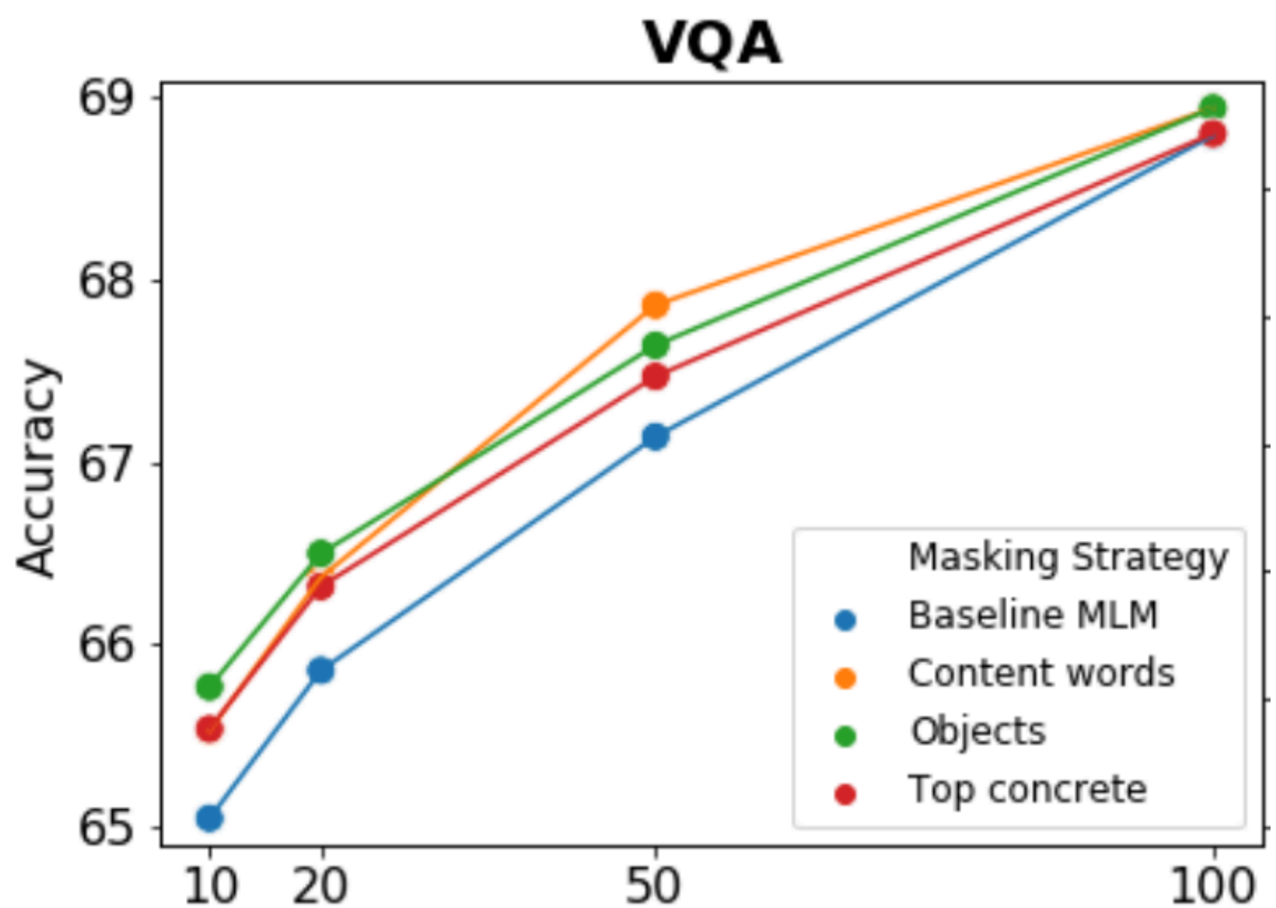
- Virtually all vision pre-training works (Shin et al., 2021) follow BERT and randomly mask 15% of the tokens
- Of the masked tokens, roughly one half are *stop-words* or *punctuation*
- We propose better masking strategies for V&L MLM

MLM in Vision and Language

- Virtually all vision pre-training works (Shin et al., 2021) follow BERT and randomly mask 15% of the tokens
- Of the masked tokens, roughly one half are *stop-words* or *punctuation*
- We propose better masking strategies for V&L MLM
- *See Yonatan's talk for more details!*

Improved Downstream Performance

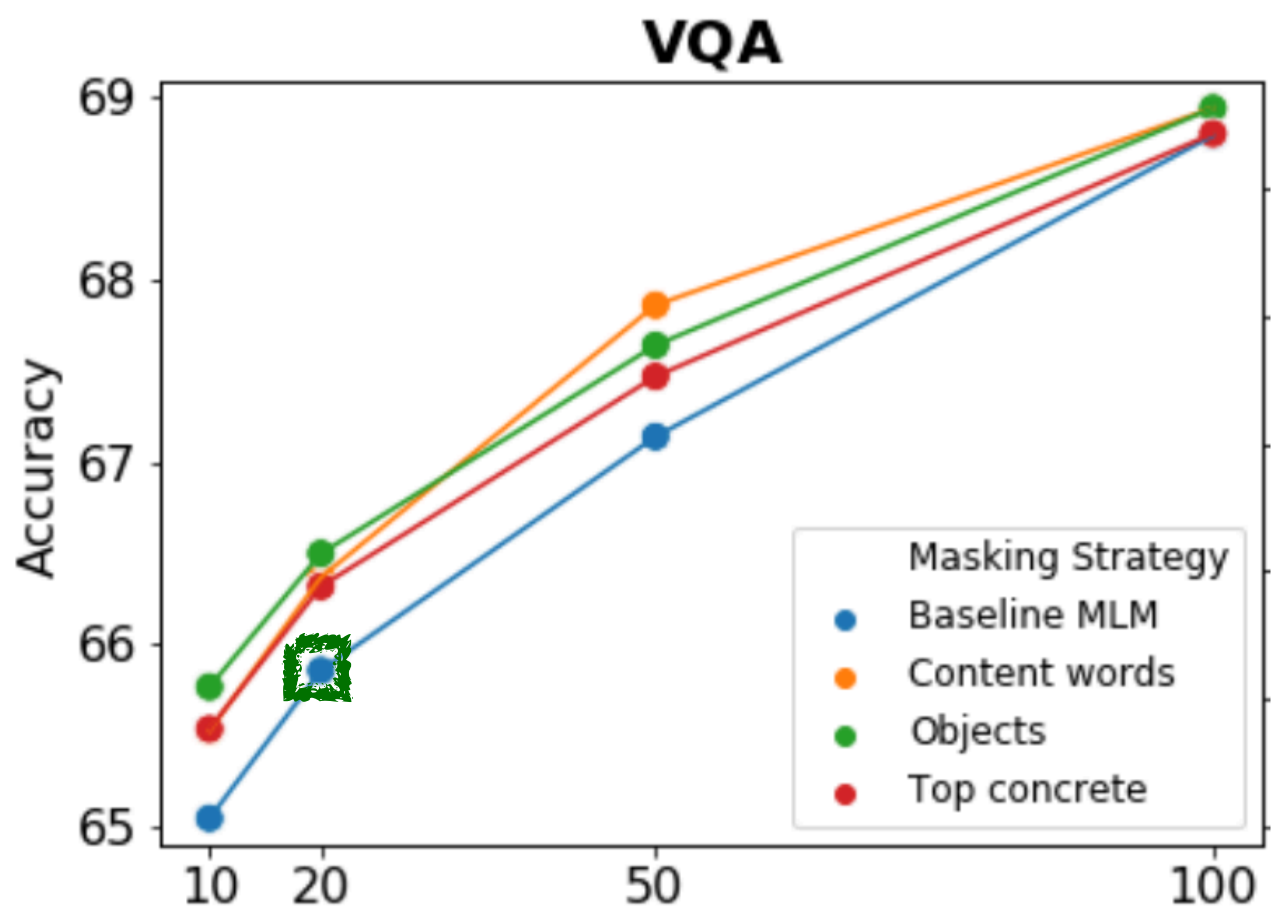
Especially on **Low-Resource Settings**



% of training data

Improved Downstream Performance

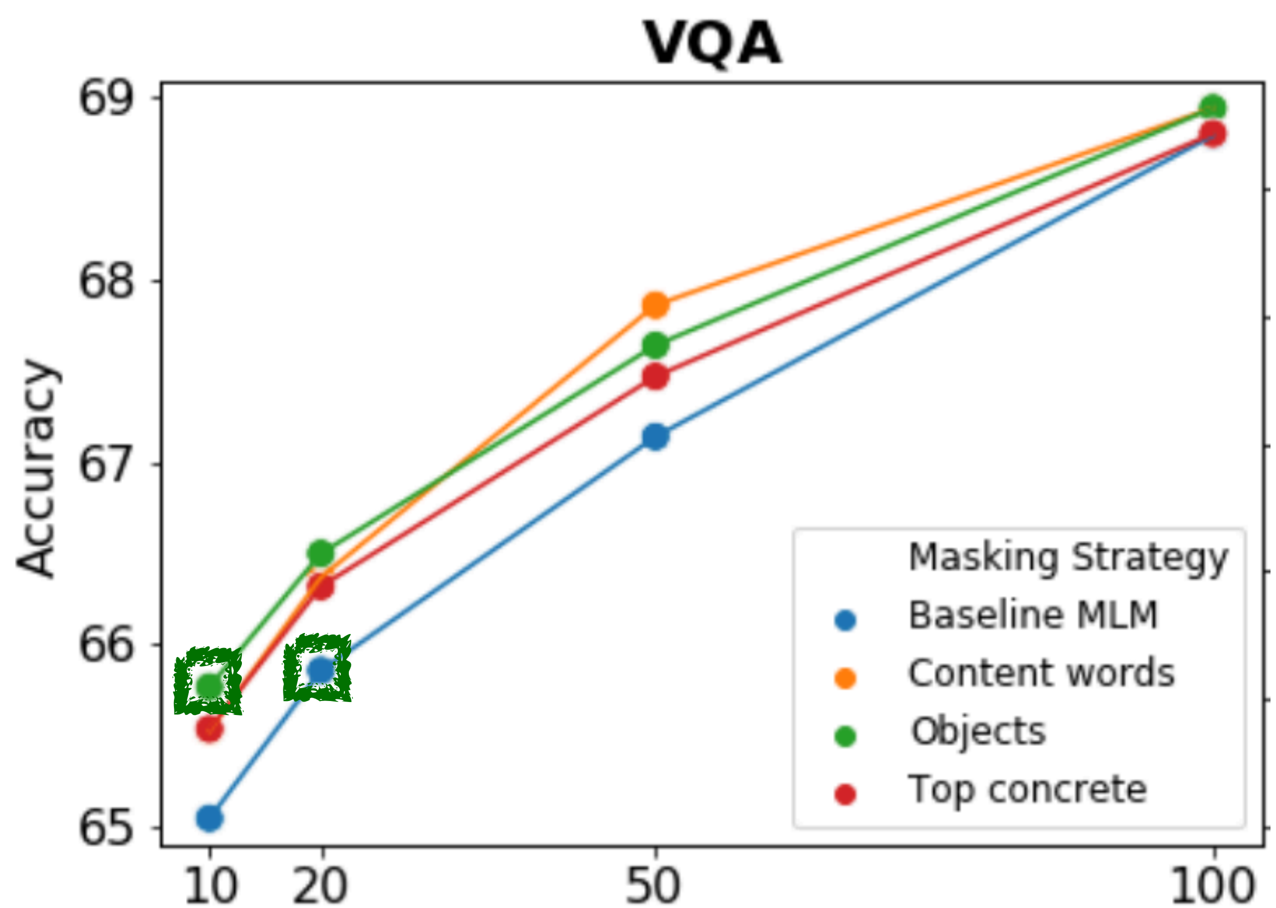
Especially on **Low-Resource Settings**



% of training data

Improved Downstream Performance

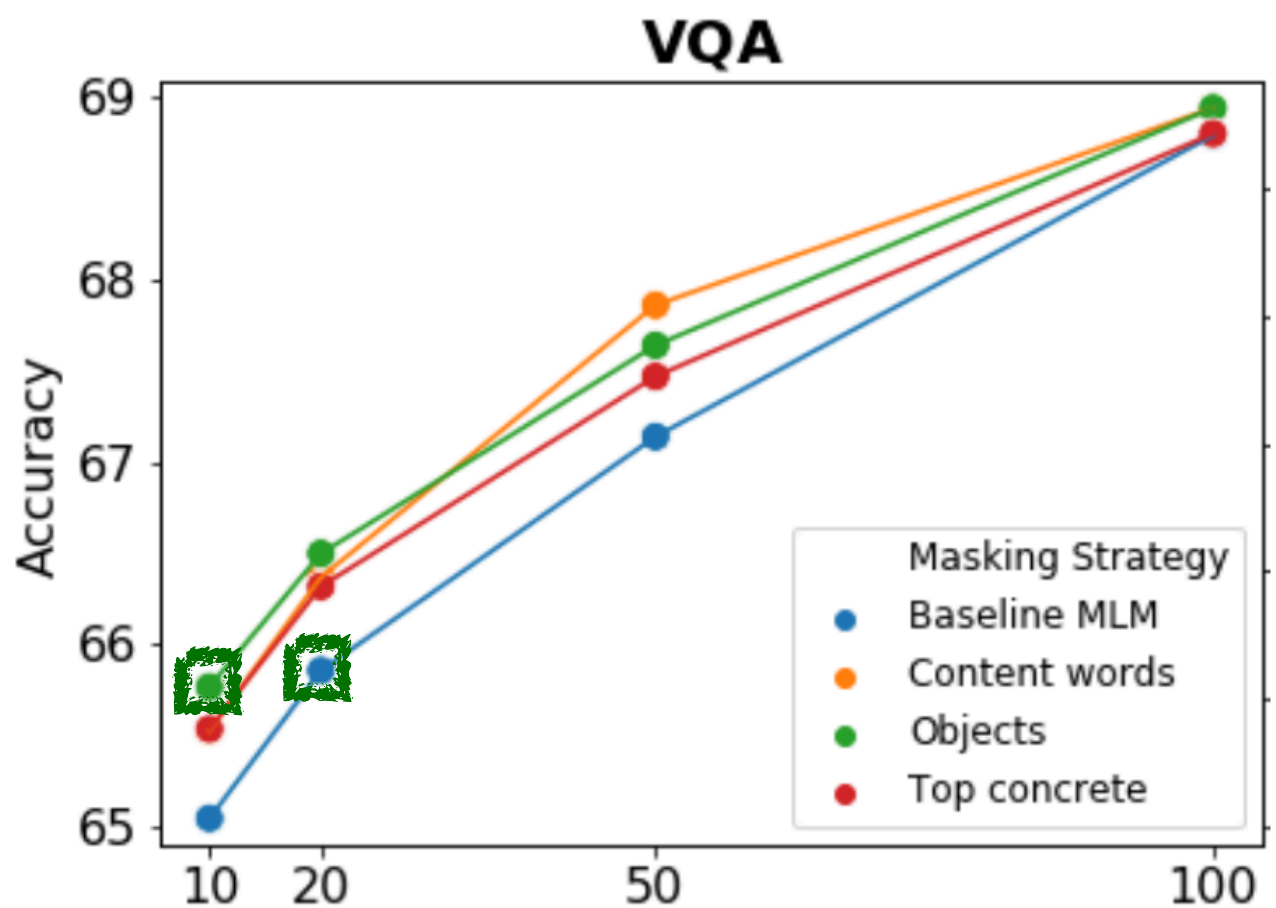
Especially on **Low-Resource Settings**



% of training data

Improved Downstream Performance

Especially on **Low-Resource Settings**

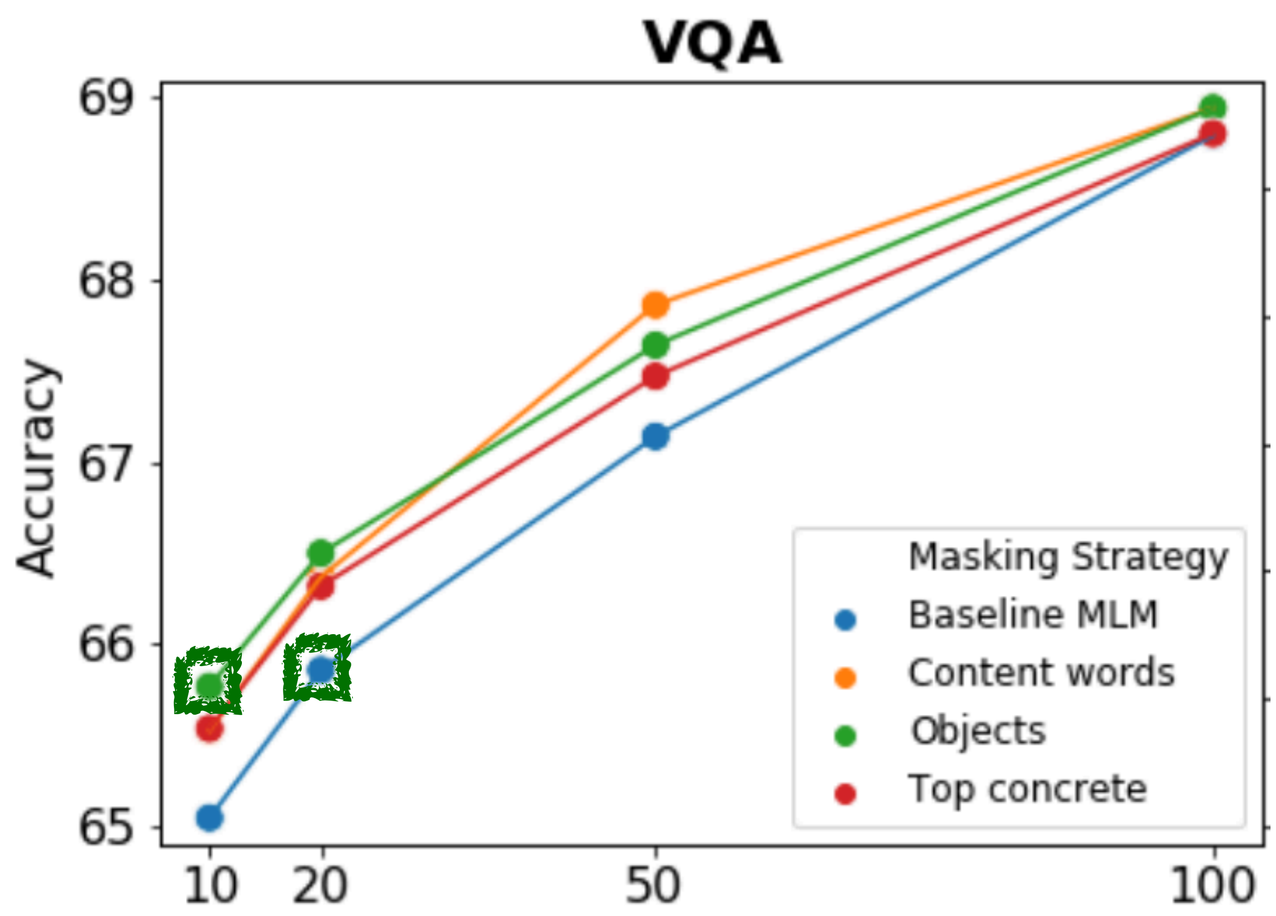


% of training data

Similar accuracy, twice as fast

Improved Downstream Performance

Especially on **Low-Resource Settings**



% of training data

Similar results on GQA, NLVR2

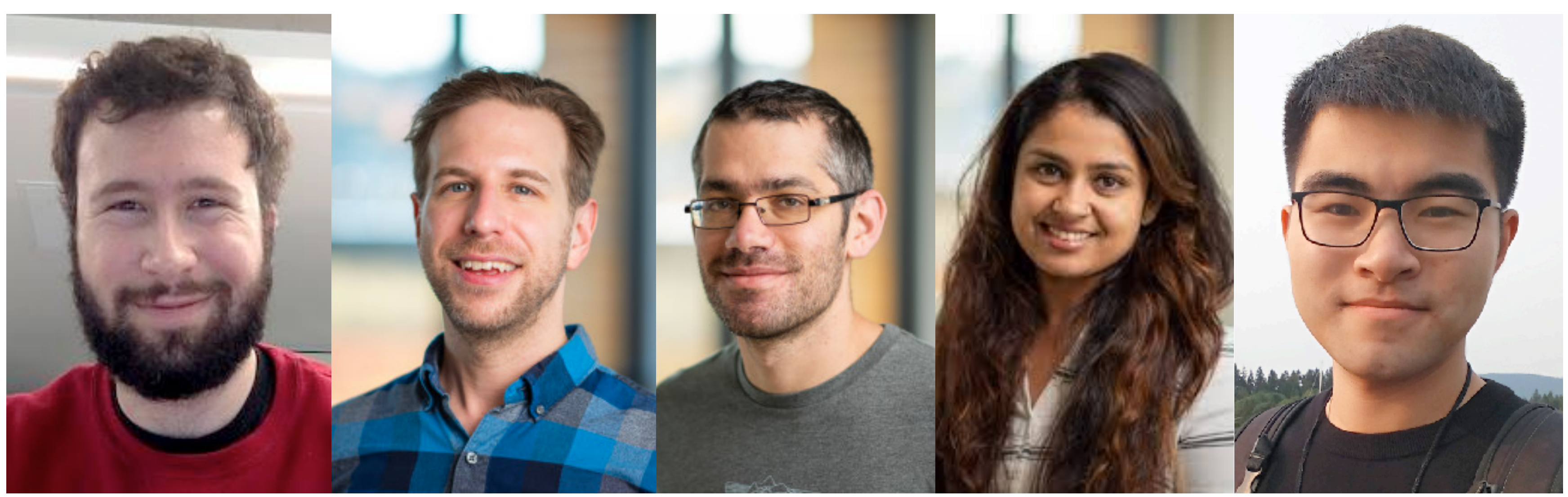
Similar accuracy, twice as fast

Not all Instances are Alike

Recap

- Efficient **inference** by selecting *the right tool for the job*
- Efficient **fine-tuning** by selecting the most *ambiguous* examples
- Efficient multi-modal **pre-training** by better *masking strategies*

Amazing Collaborators!



Not all instances are alike

Recap

- Efficient **inference** by selecting *the right tool for the job*
- Efficient **fine-tuning** by selecting the most *ambiguous* examples
- Efficient multi-modal **pre-training** by better *masking strategies*