

Towards Anytime Uncertainty Estimation in Early-Exit Neural Networks

Eric Nalisnick

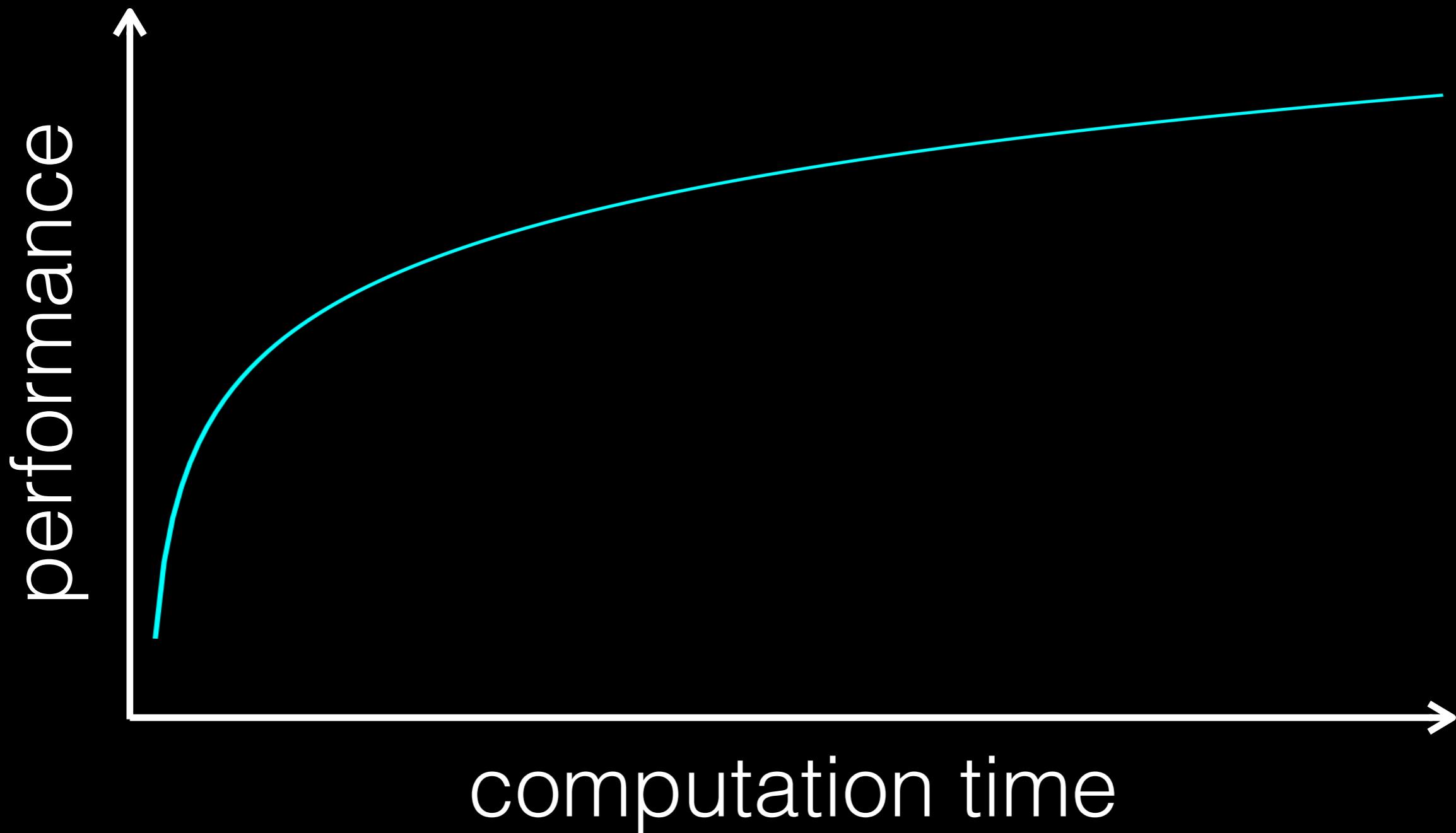
University of Amsterdam



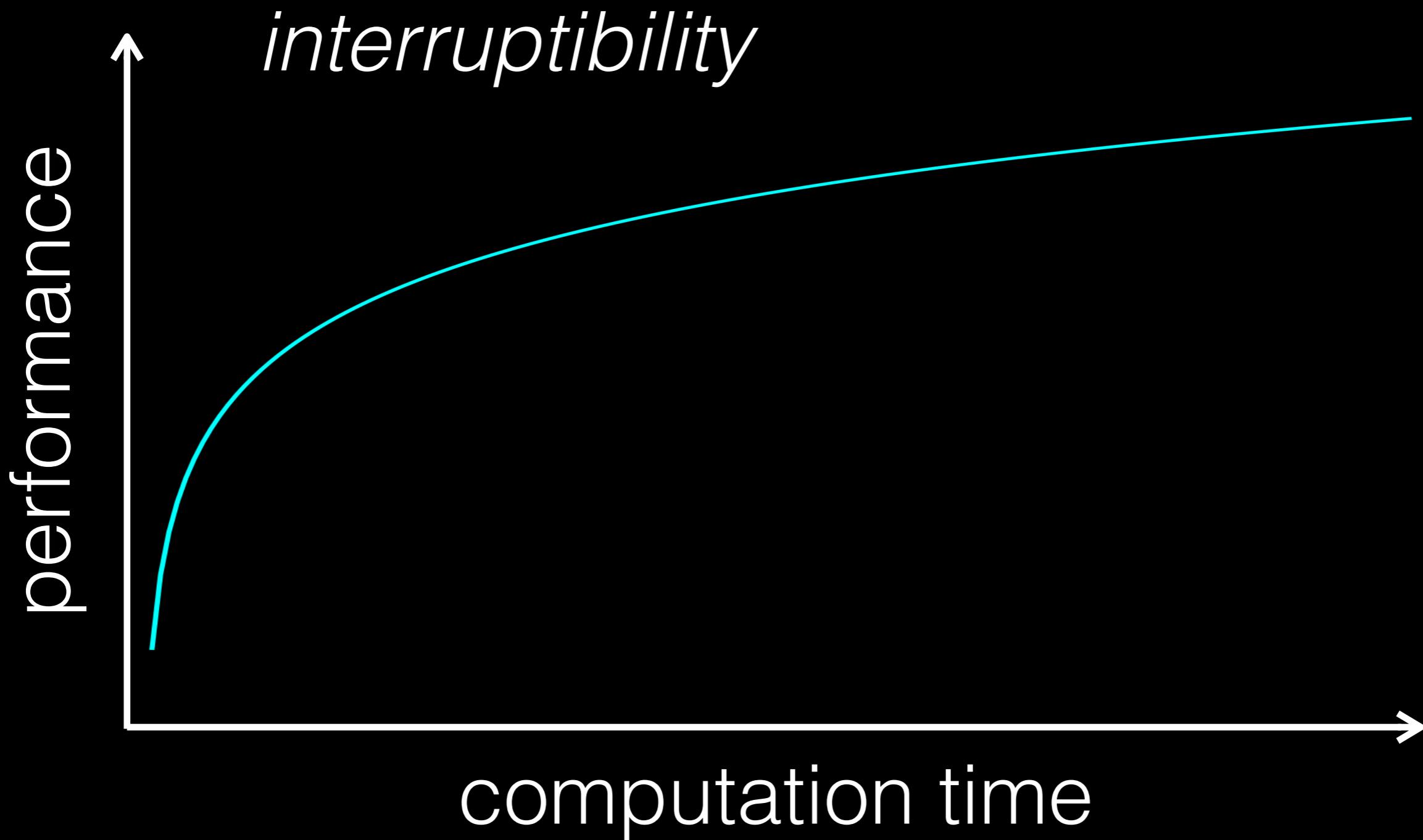


© AdobeStock/Scharfsinn86

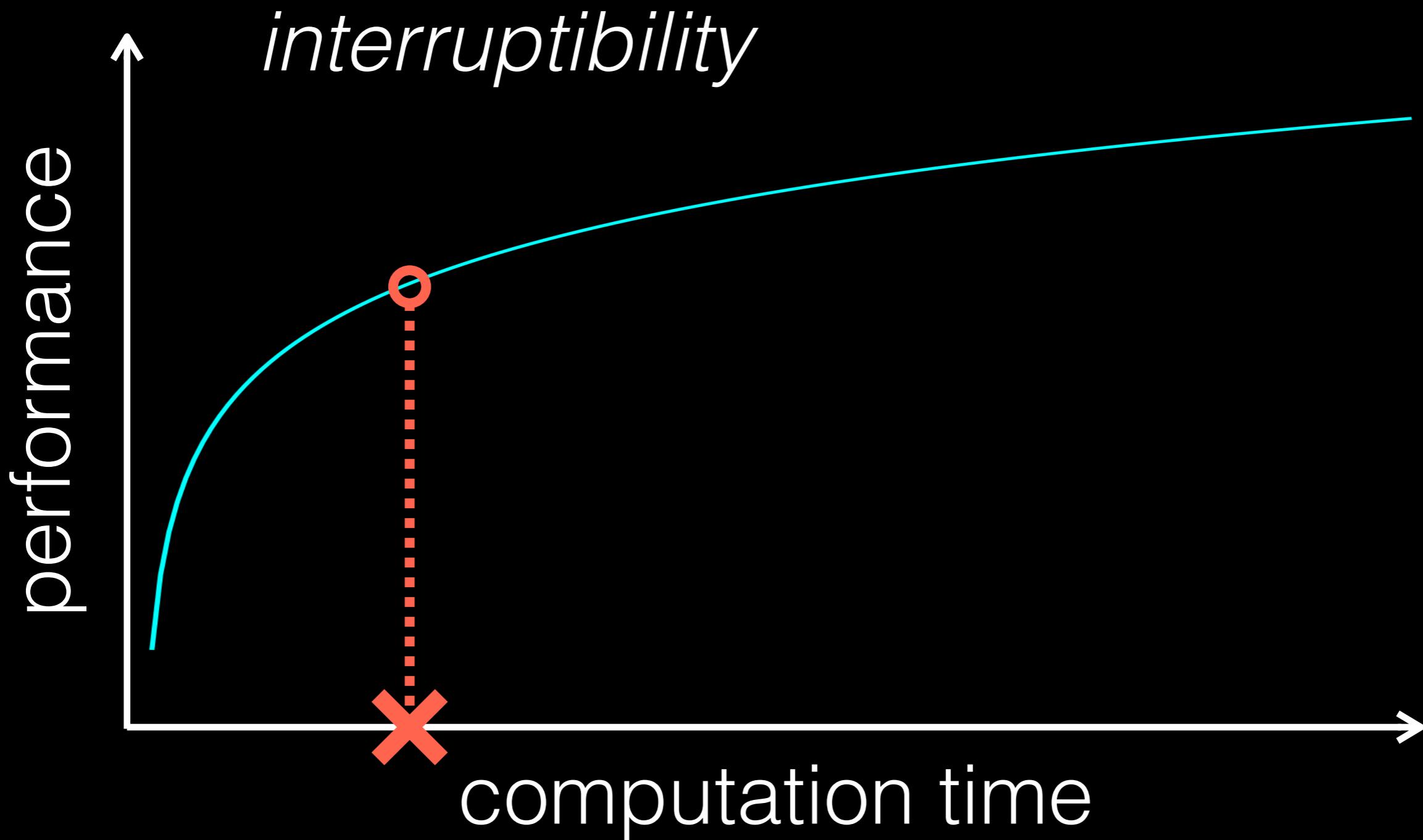
Anytime Models



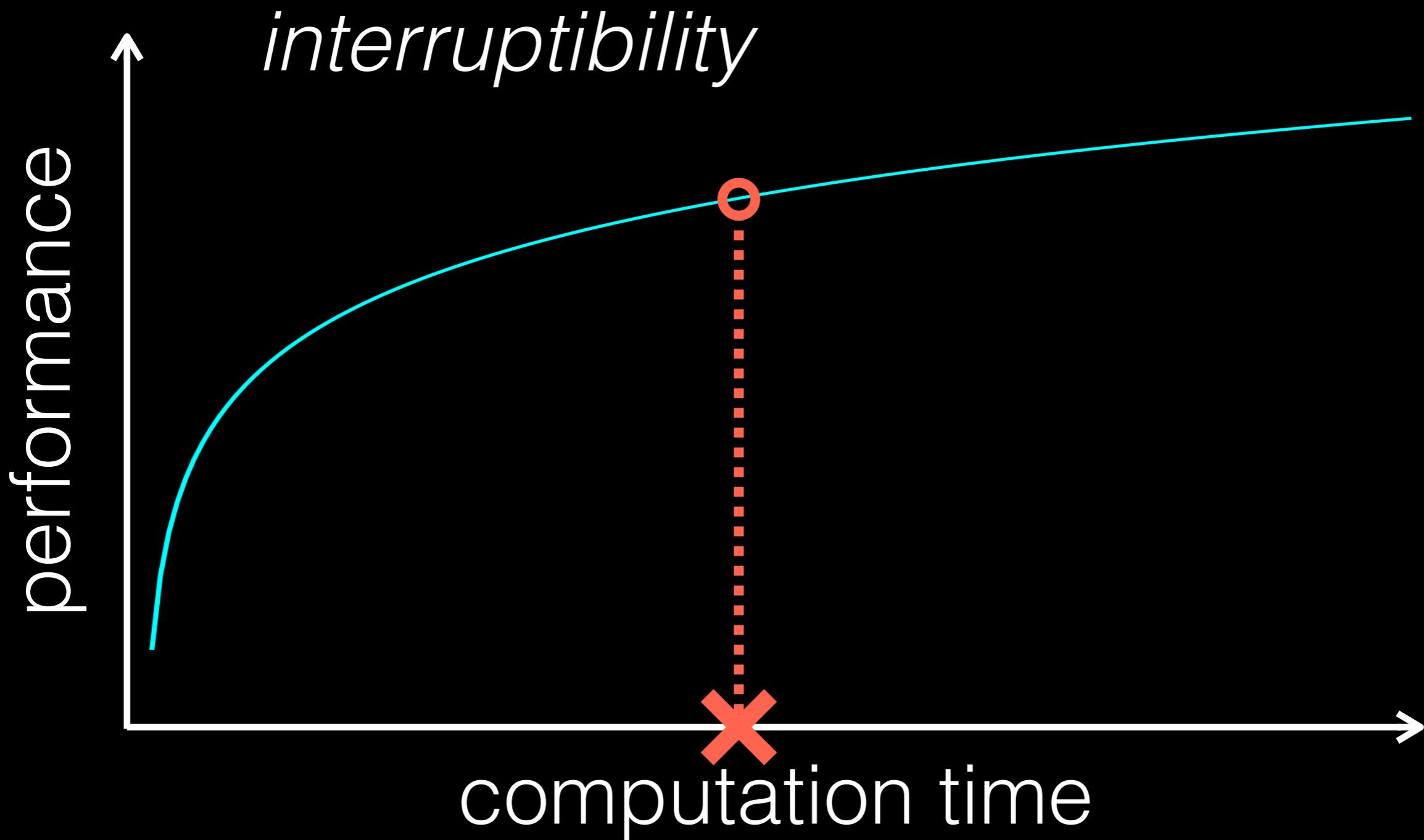
Anytime Models



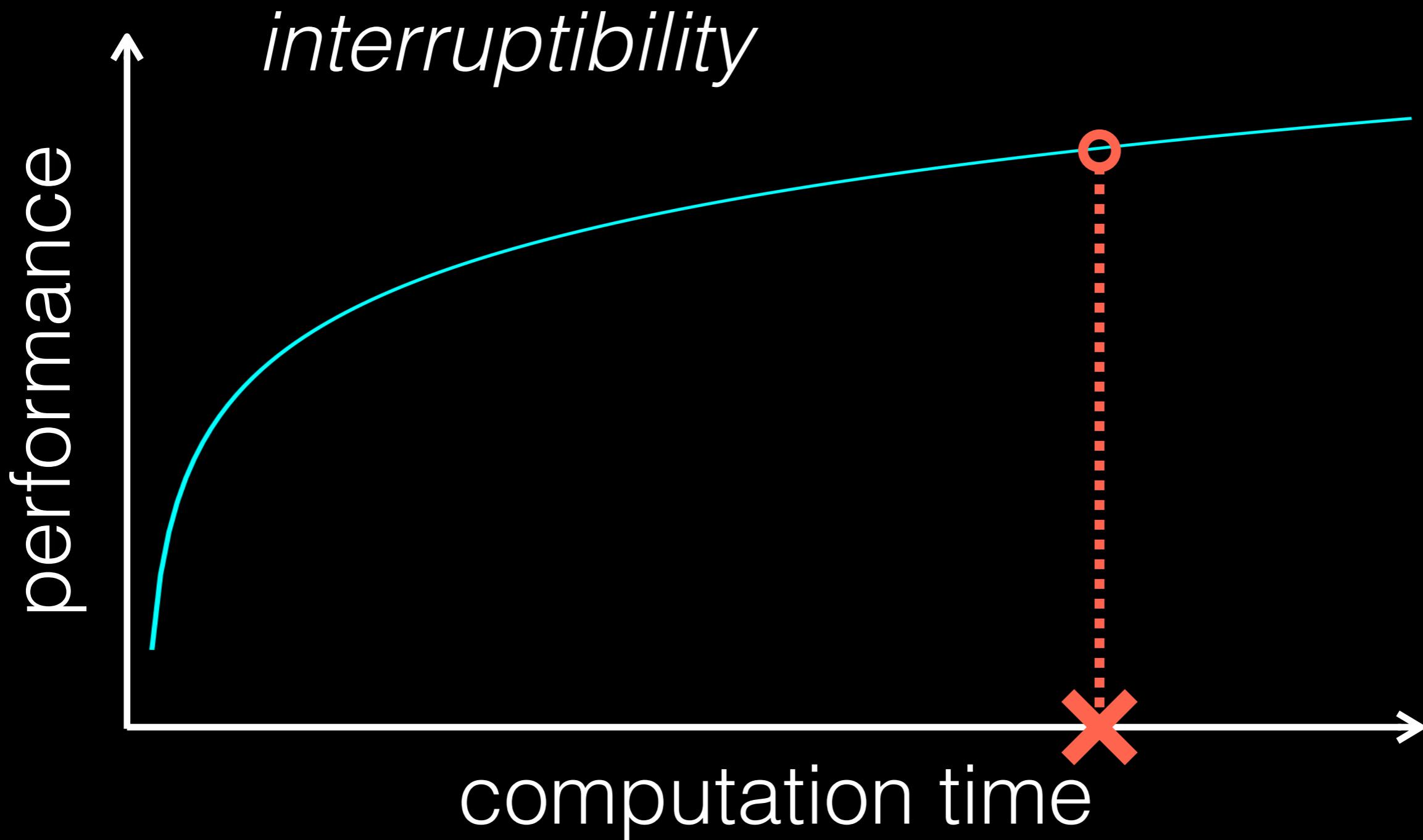
Anytime Models



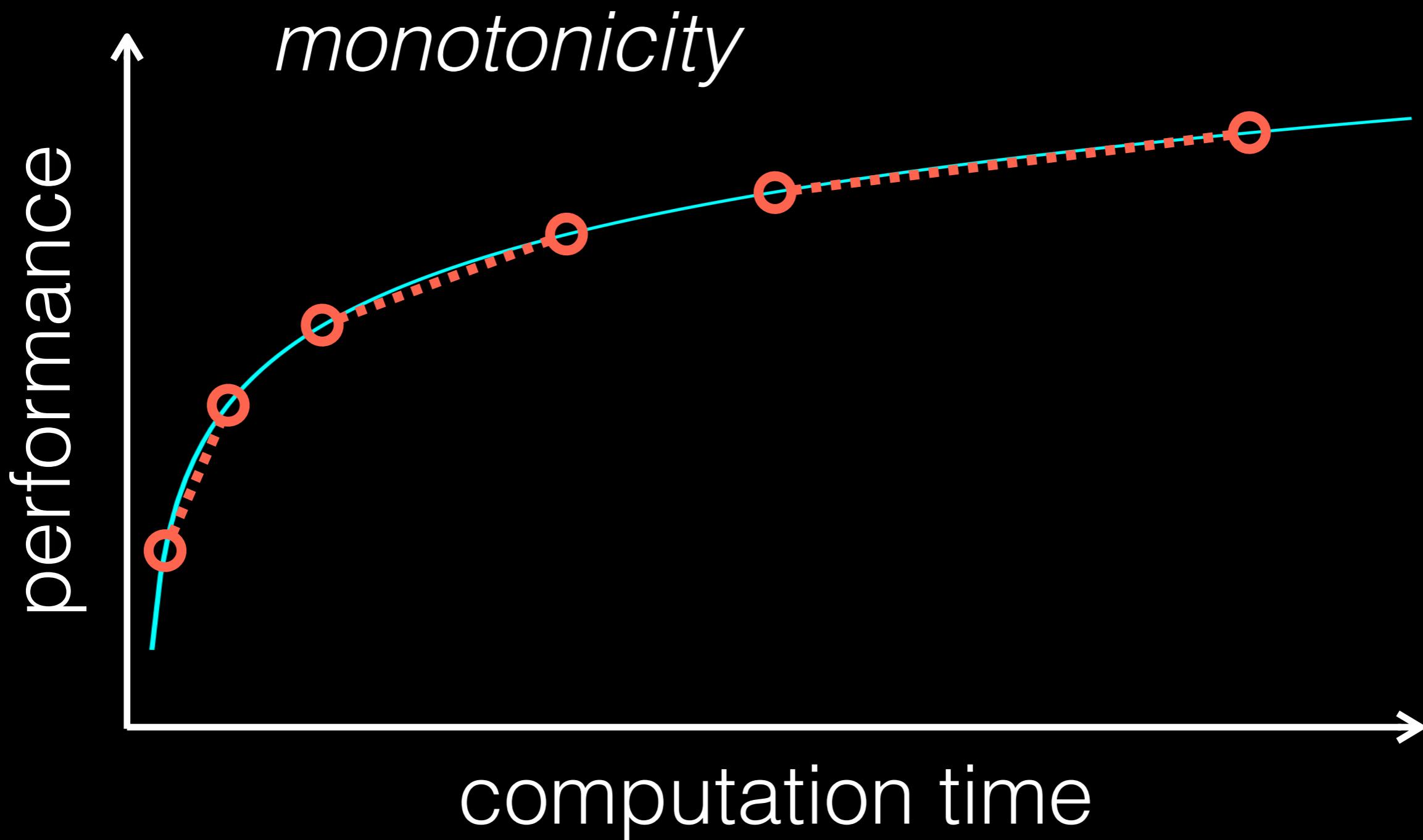
Anytime Models



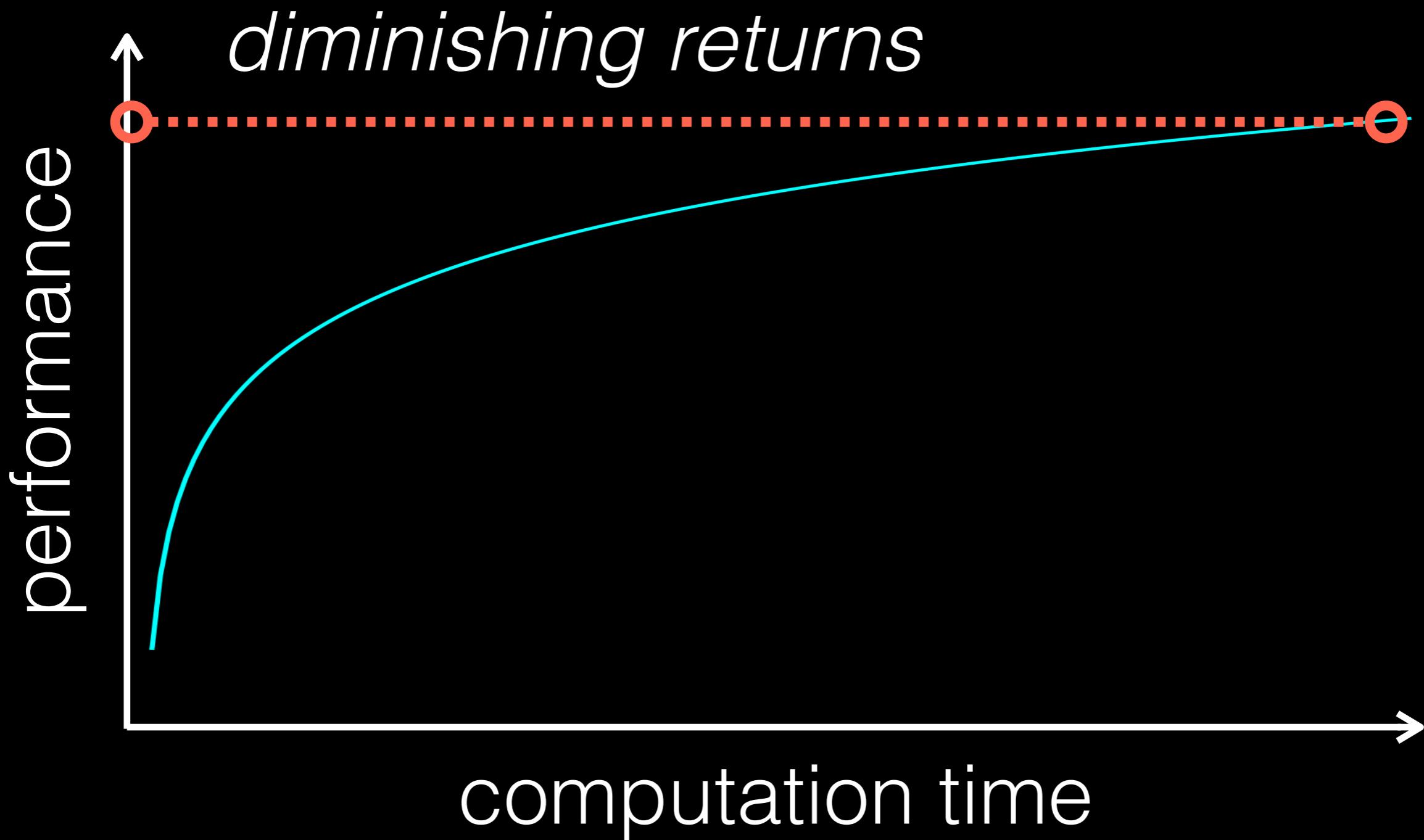
Anytime Models



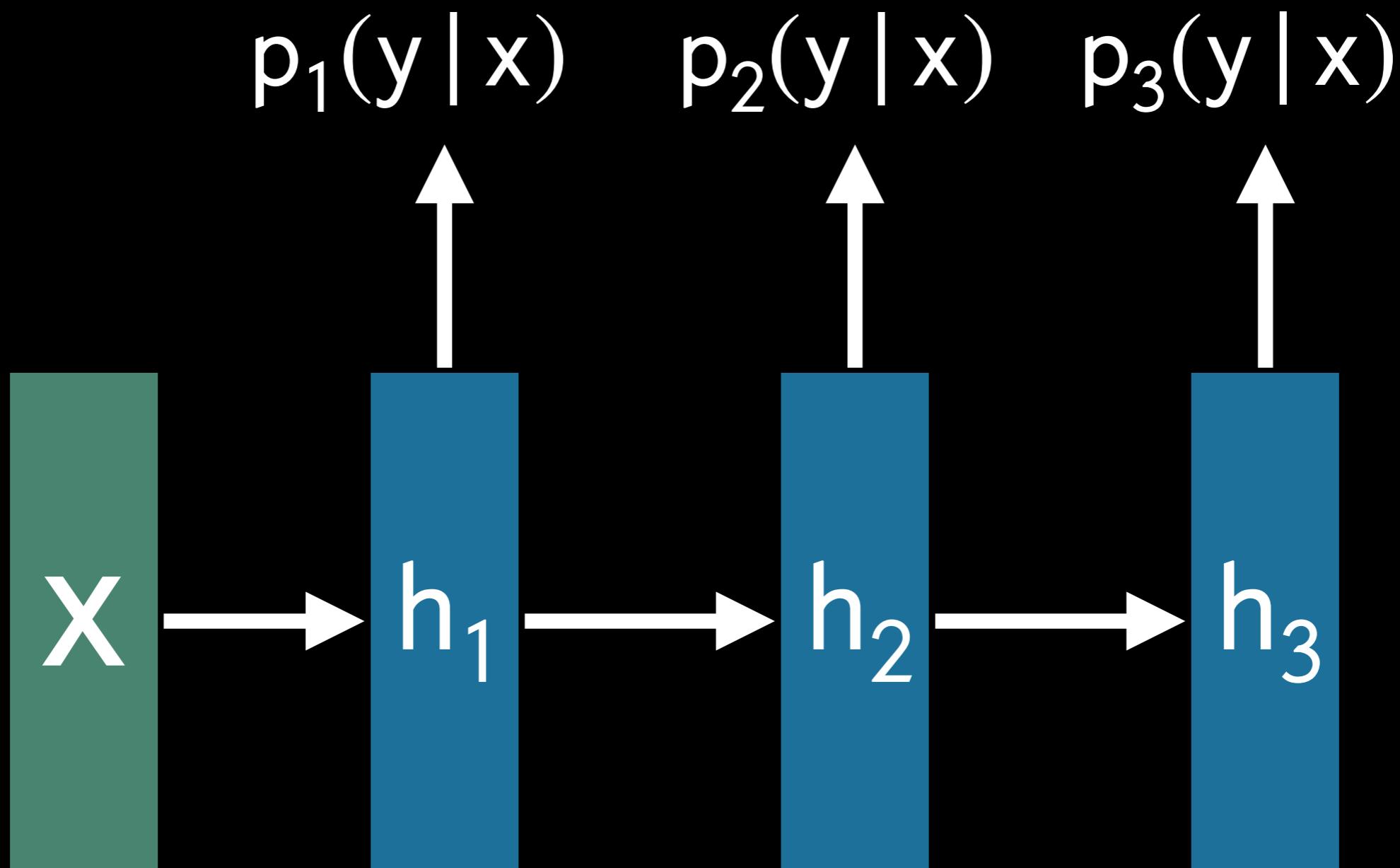
Anytime Models



Anytime Models

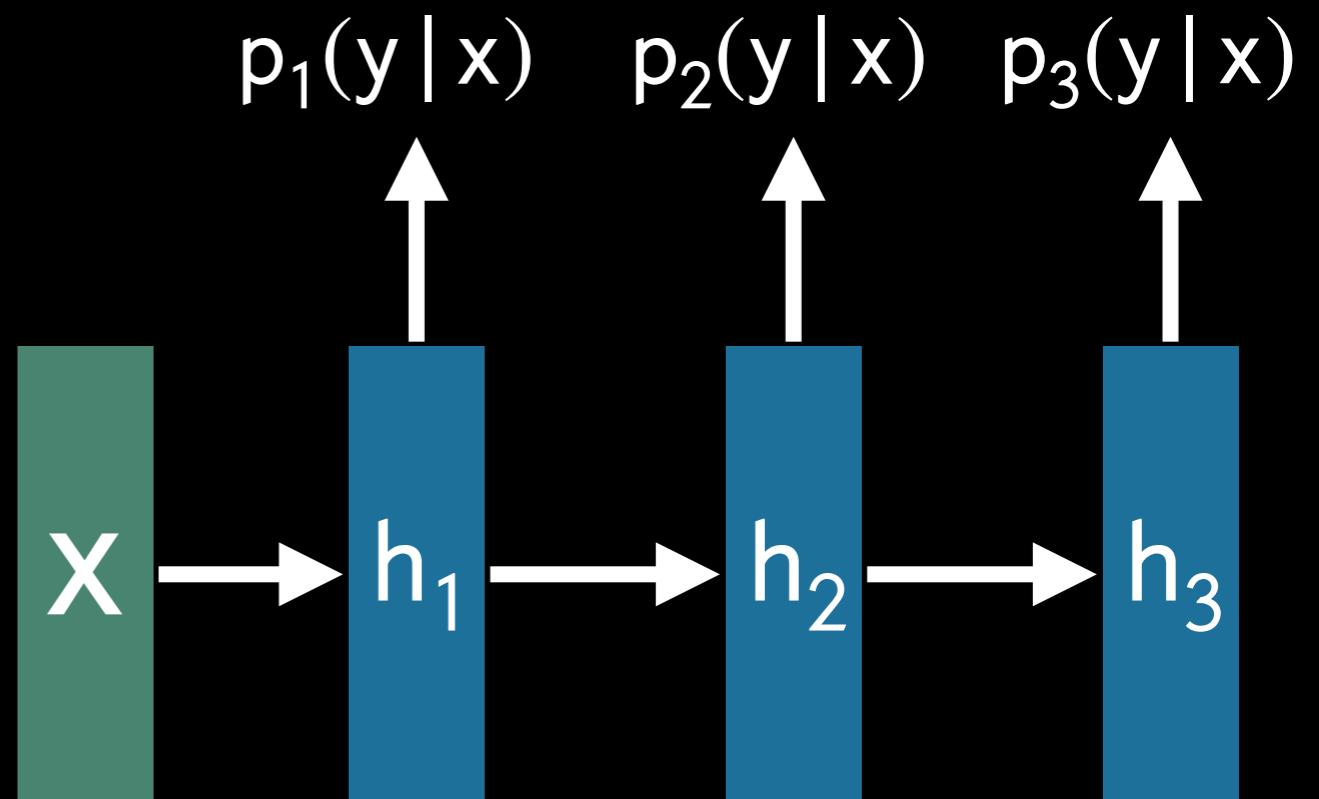


Early-Exit Neural Networks



Early-Exit Neural Networks

$$\ell(\theta_{1:E}) = - \sum_{e=1}^E \log p_e(y | x, \theta_{1:e})$$



Early-Exit Neural Networks

- ⊗ interruptibility?
- ⊗ monotonicity?
- ⊗ diminishing returns?

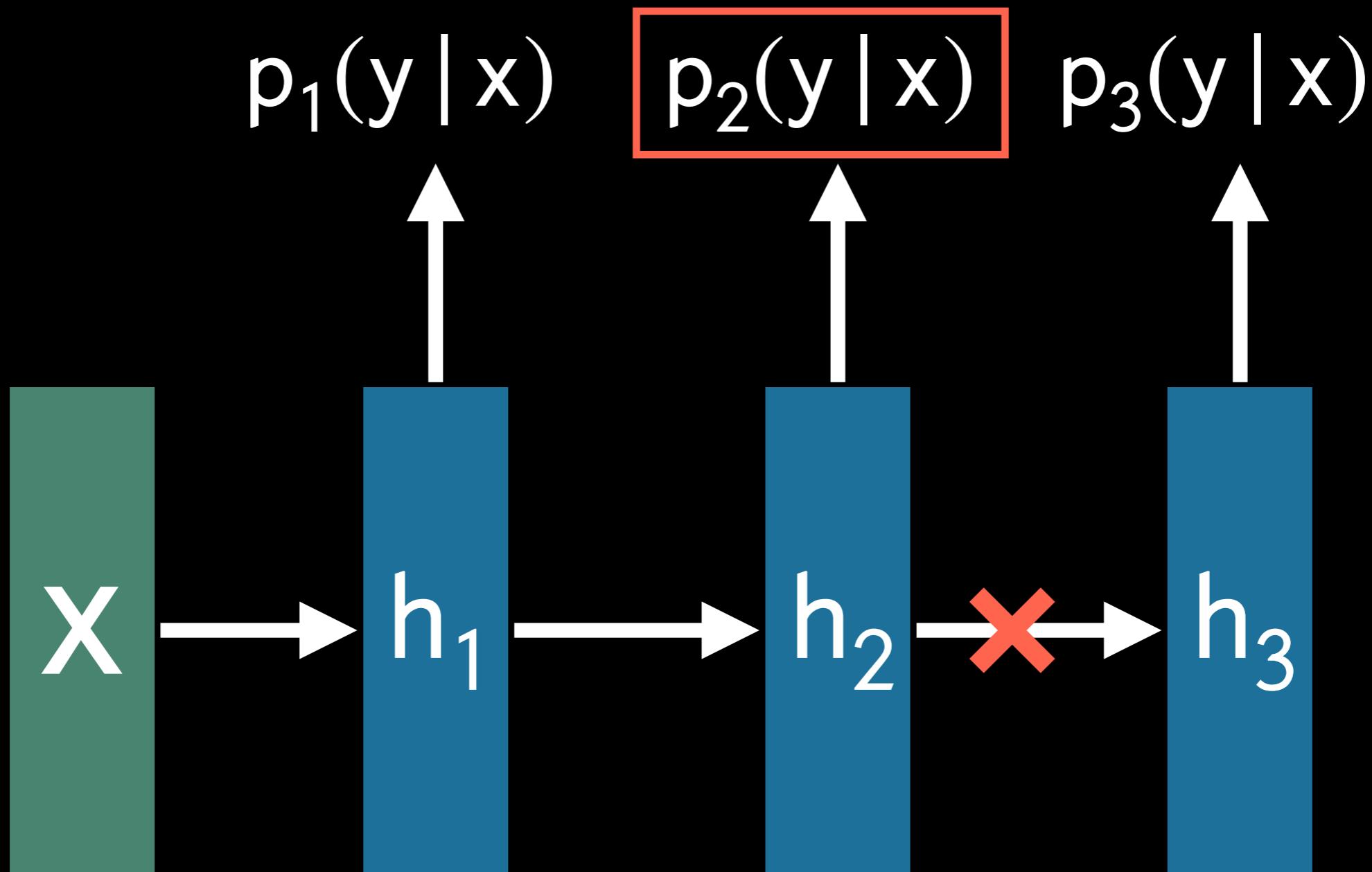
[Zilberstein, AI Magazine 1996]

Early-Exit Neural Networks

- ⊗ interruptibility?
- ⊗ monotonicity?
- ⊗ diminishing returns?

[Zilberstein, AI Magazine 1996]

Early-Exit Neural Networks



Early-Exit Neural Networks

- ⊗ interruptibility



- ⊗ monotonicity?

- ⊗ diminishing returns?

Early-Exit Neural Networks

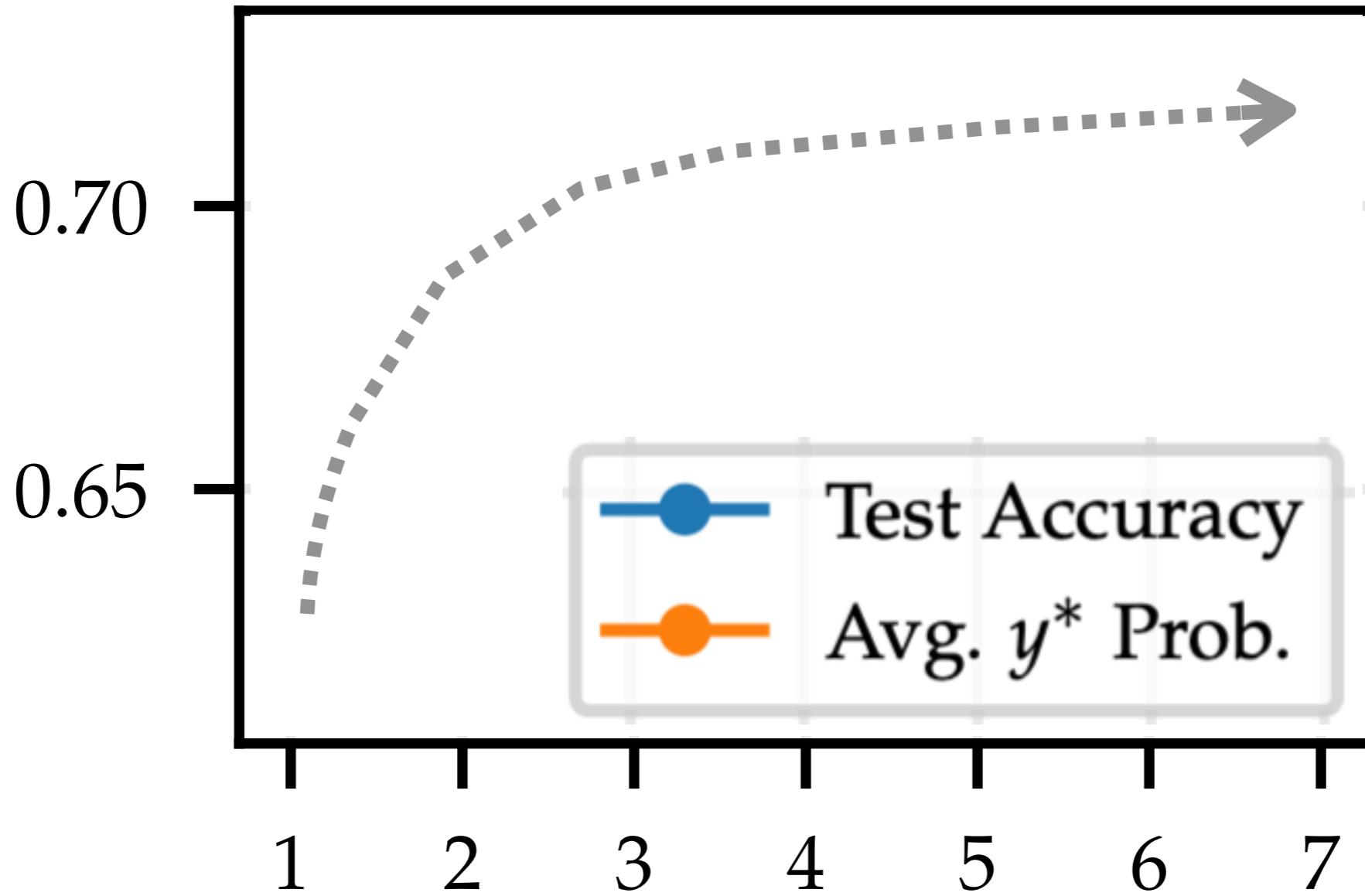
- ⊗ interruptibility



- ⊗ monotonicity?

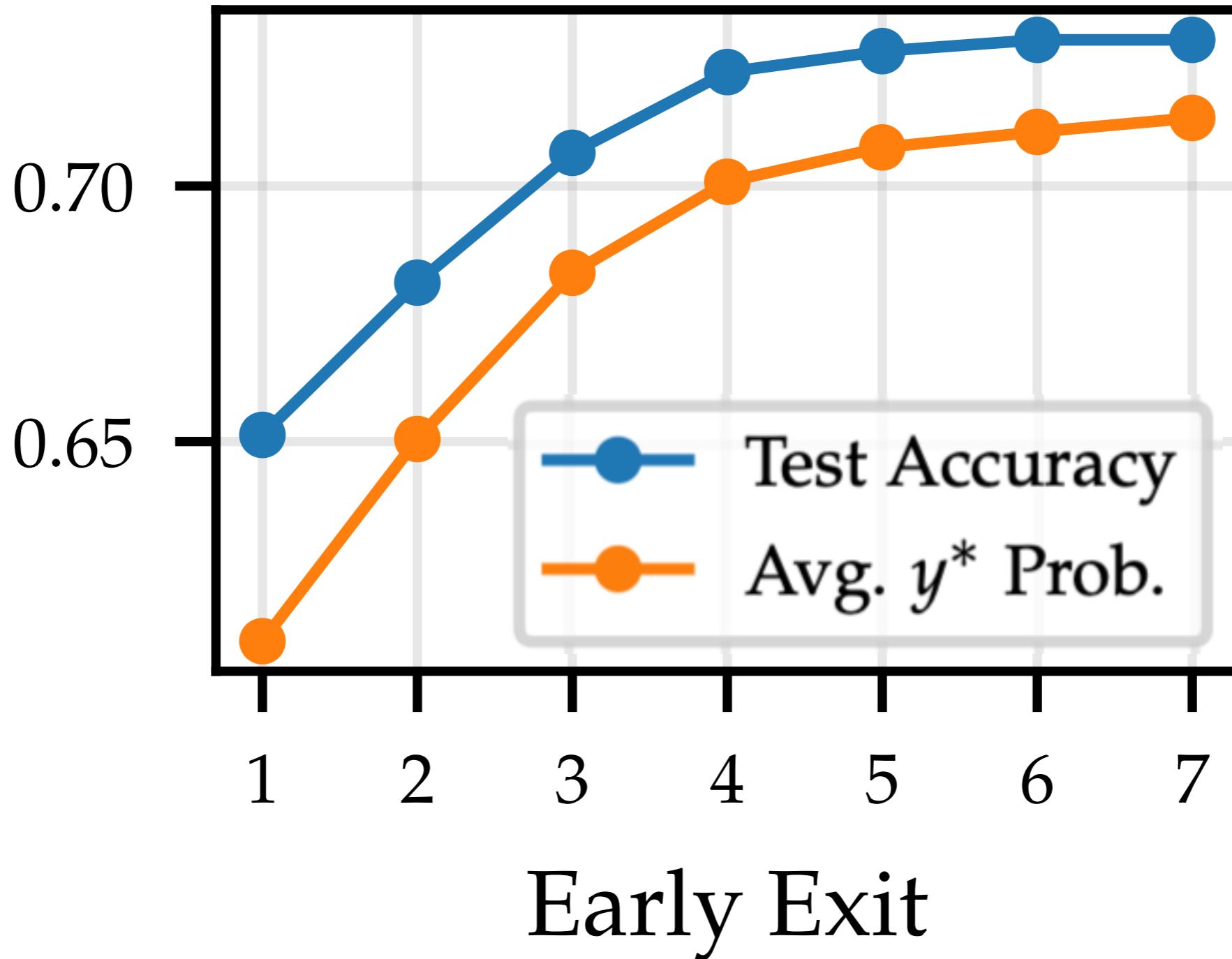
- ⊗ diminishing returns?

Multi-Scale Dense Net: CIFAR-100

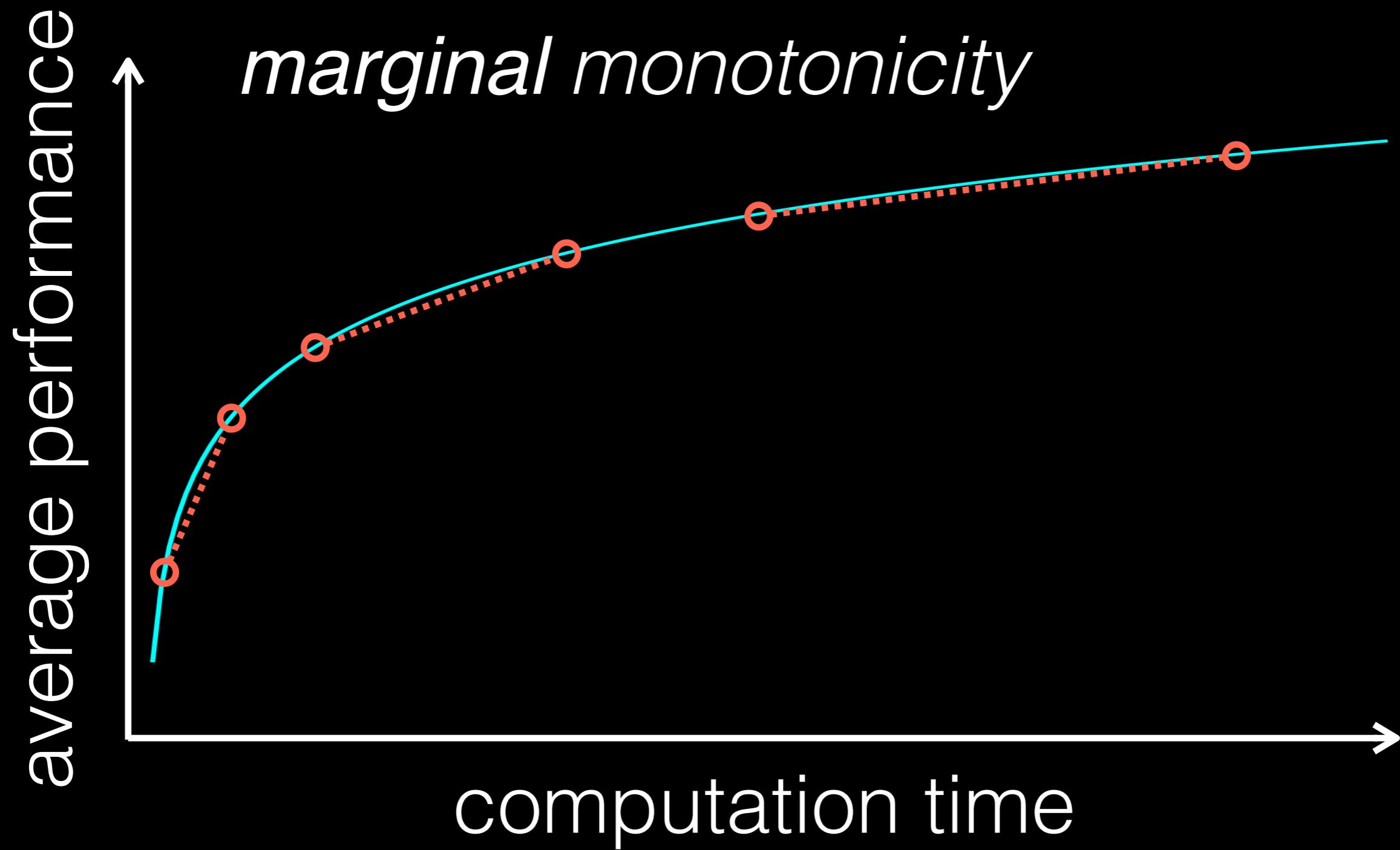


Early Exit

Multi-Scale Dense Net: CIFAR-100



Anytime Models



Early-Exit Neural Networks

⊗ interruptibility

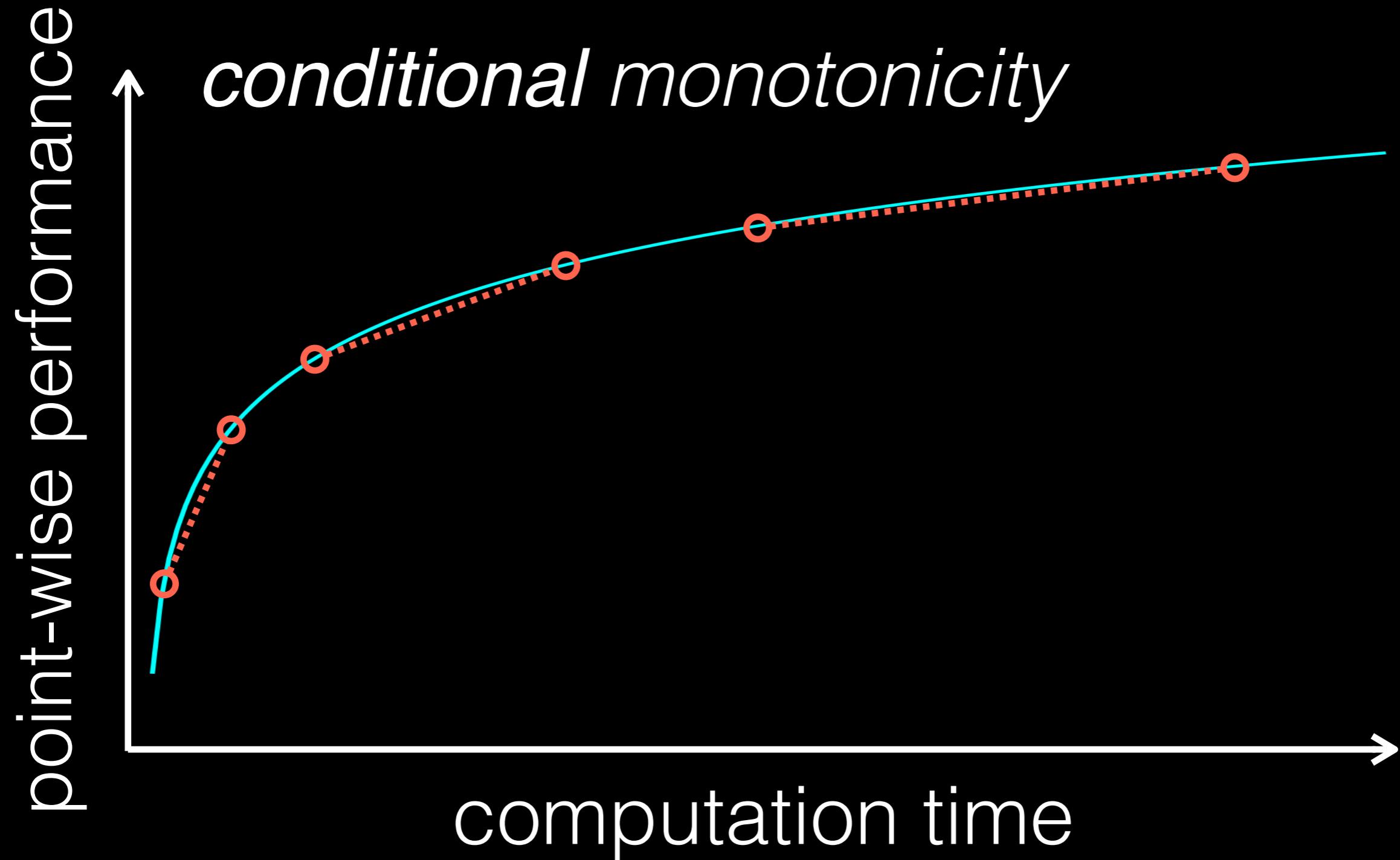


⊗ monotonicity?

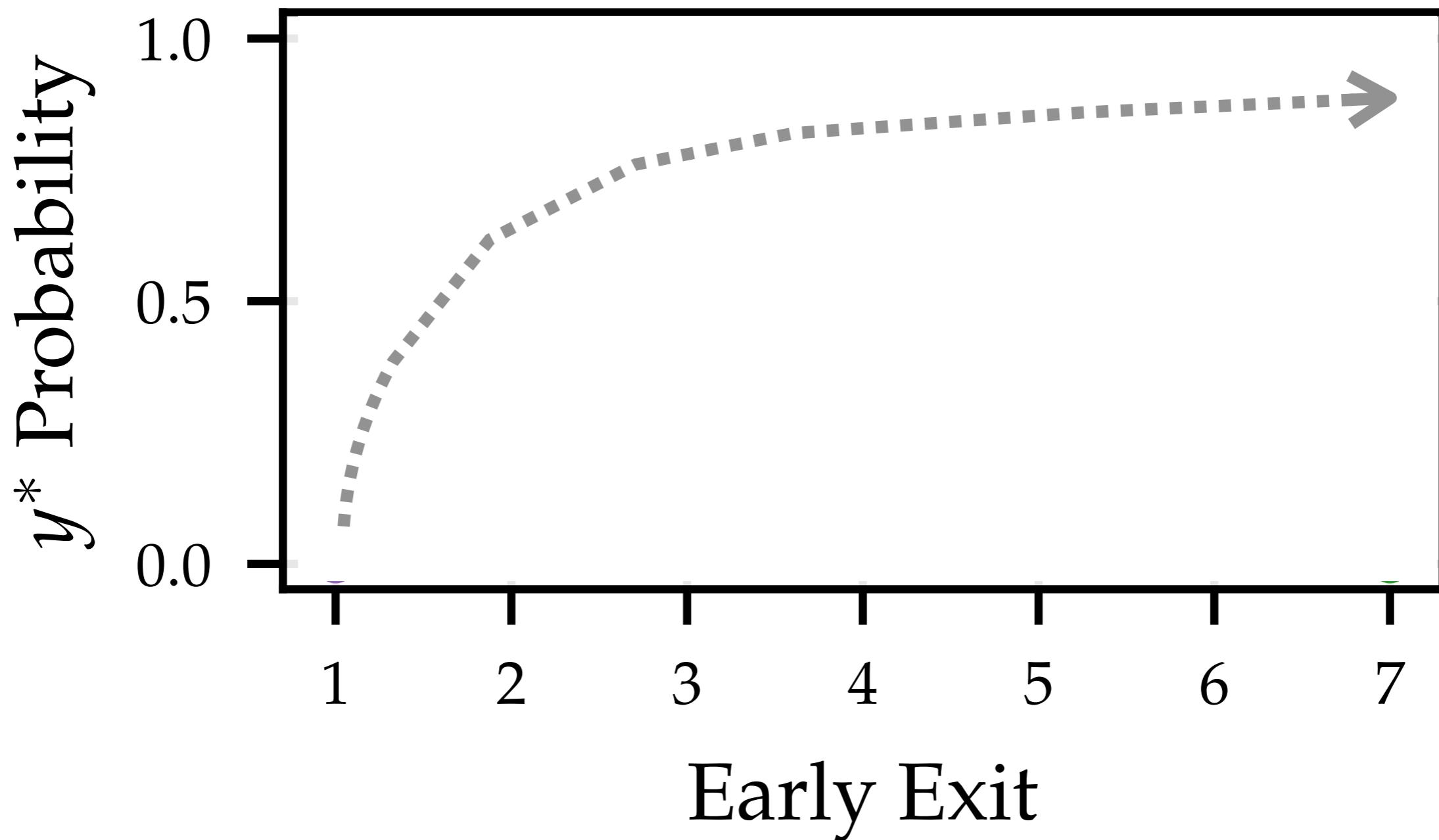


⊗ diminishing returns?

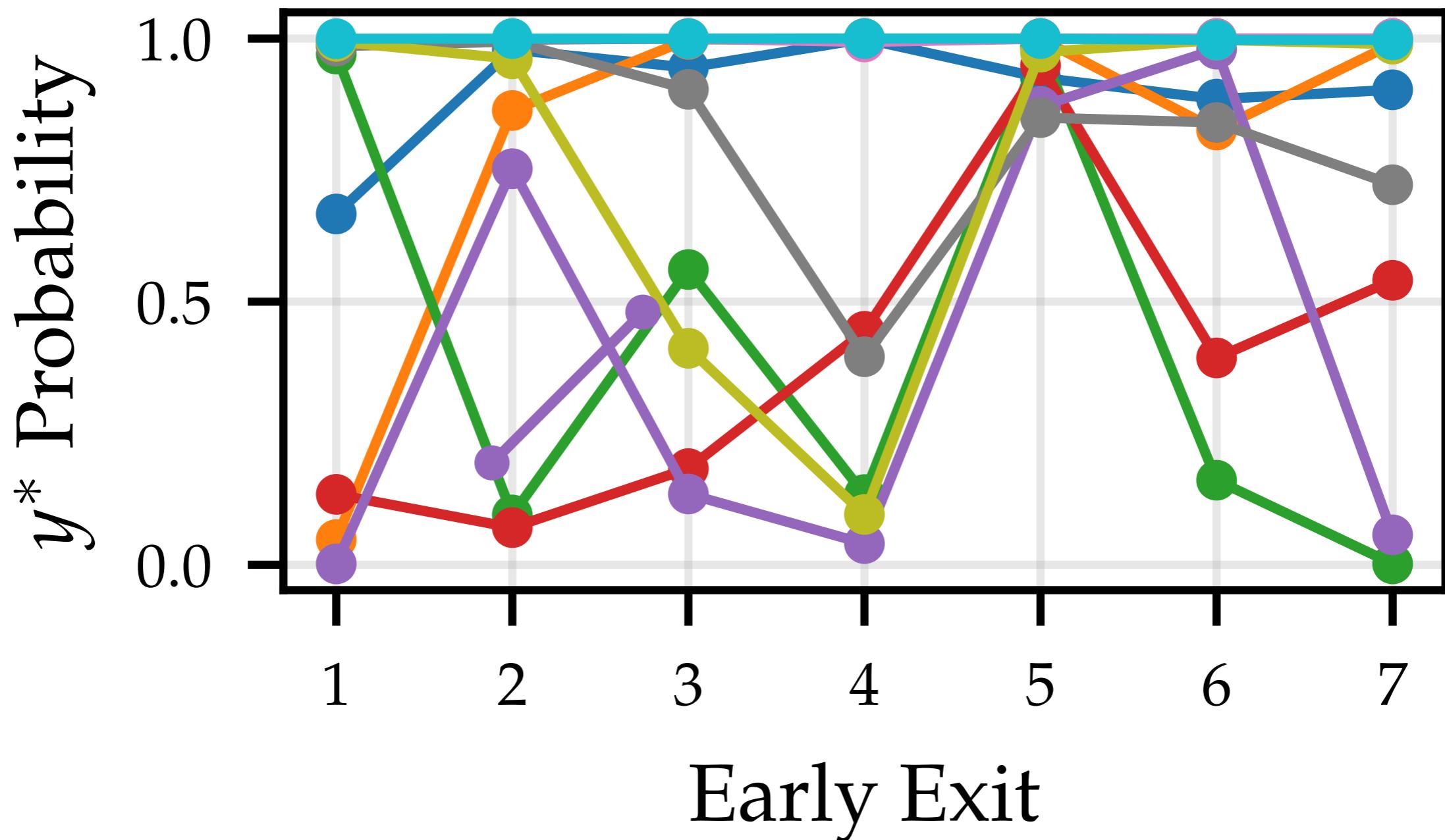
Anytime Models



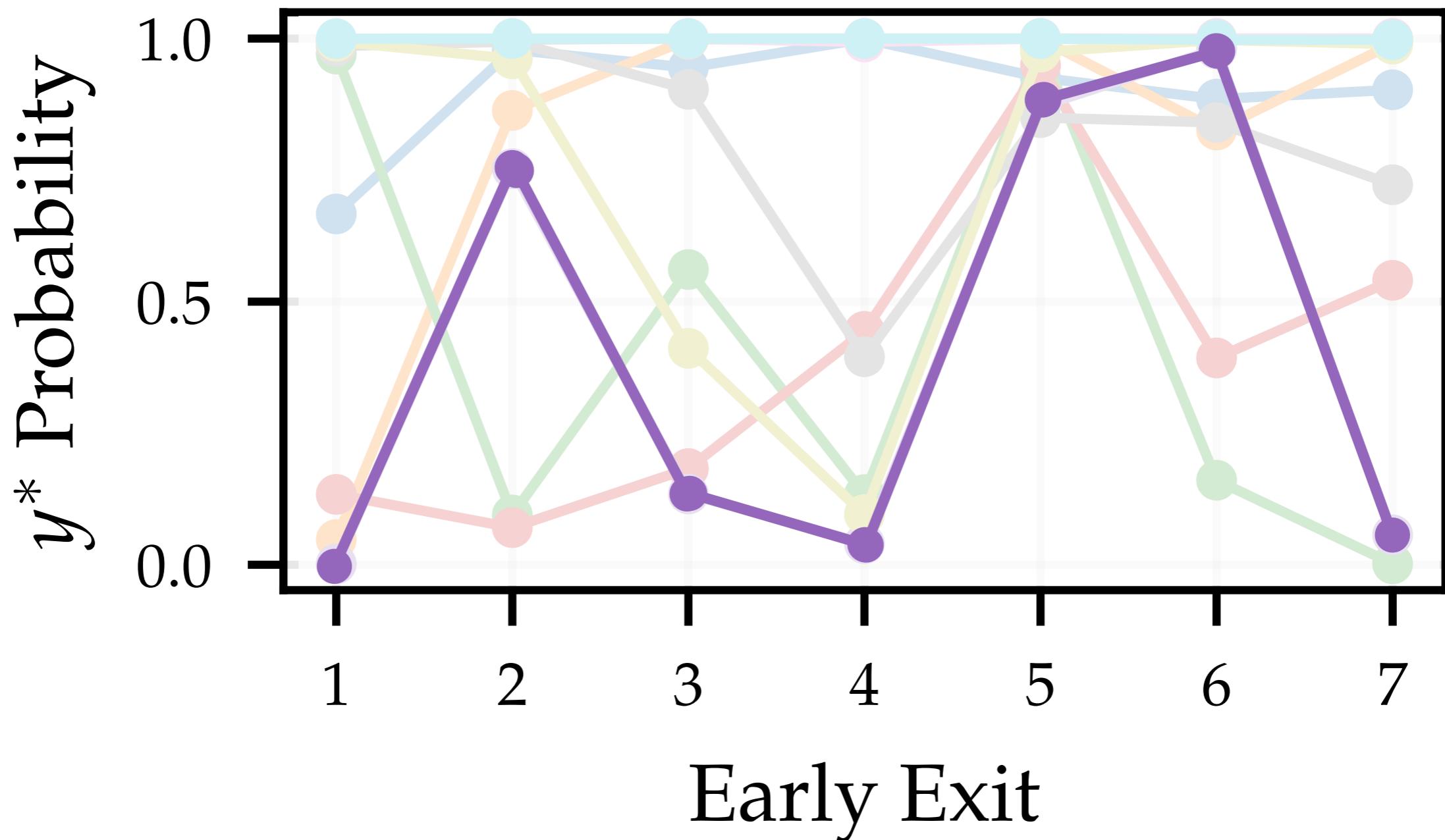
Multi-Scale Dense Net: CIFAR-100



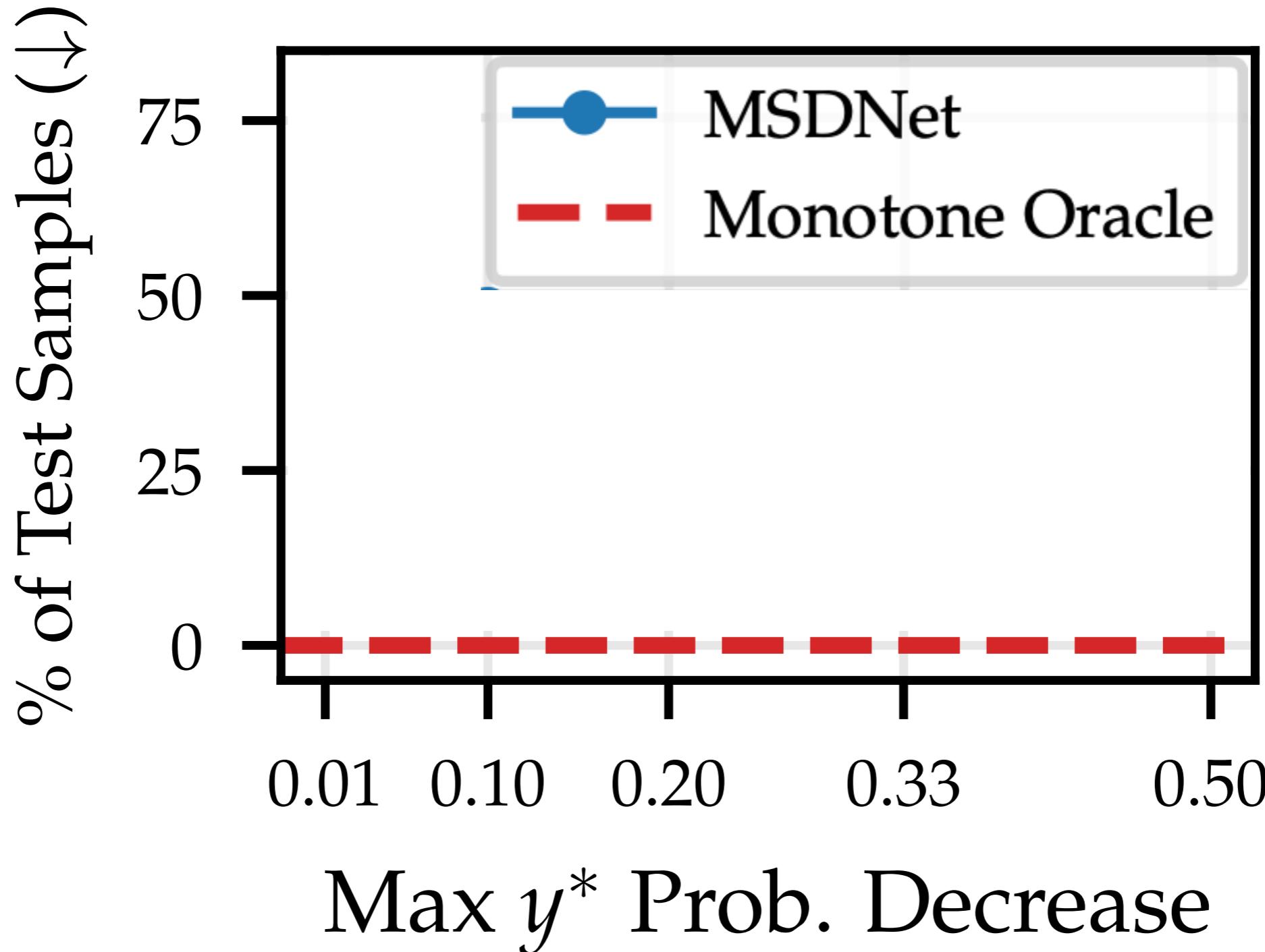
Multi-Scale Dense Net: CIFAR-100



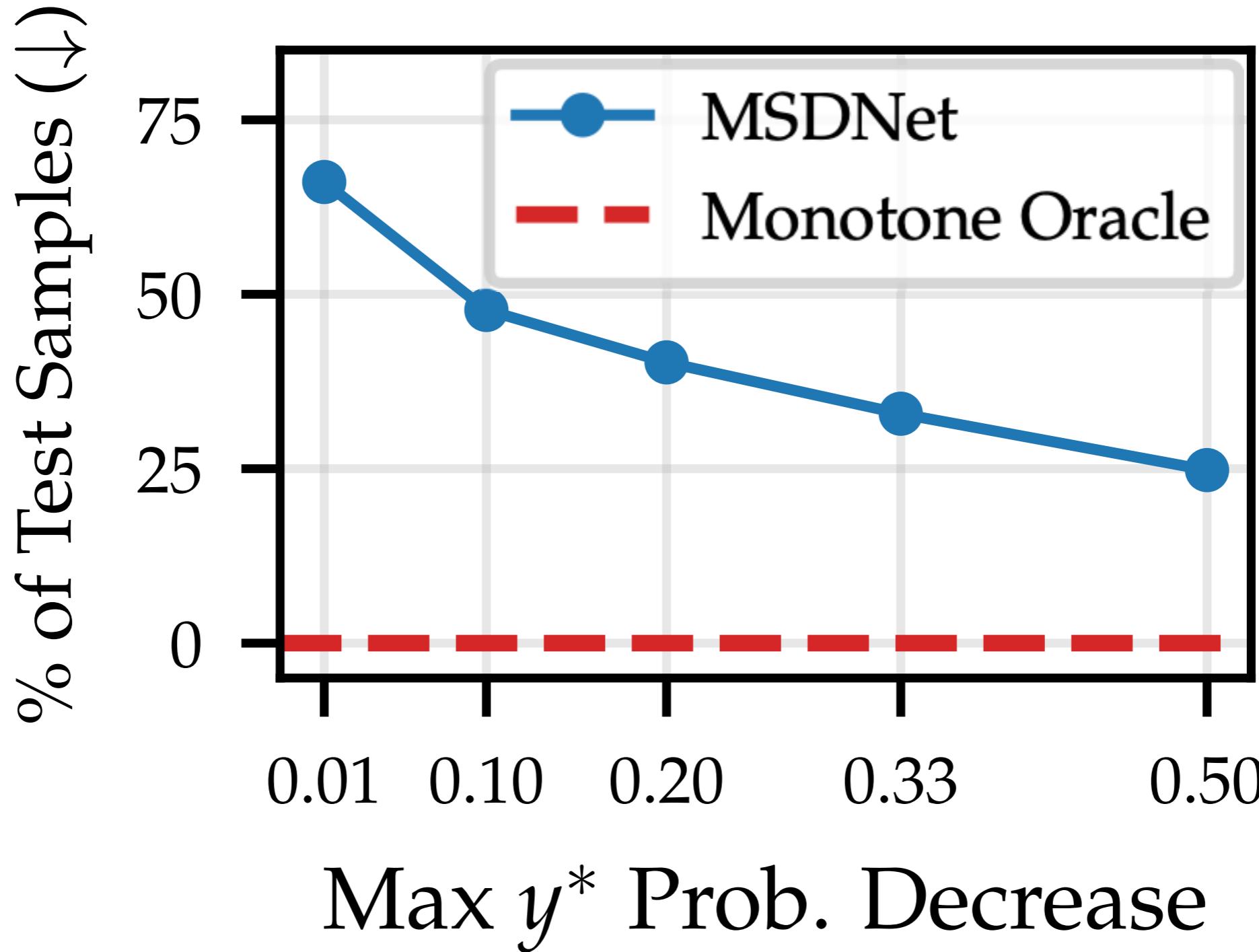
Multi-Scale Dense Net: CIFAR-100



Multi-Scale Dense Net: CIFAR-100



Multi-Scale Dense Net: CIFAR-100



Multi-Scale Dense Net: Overthinking

Overthinking: having the correct prediction but then switching to a wrong prediction.

[Kaya et al., ICML 2019]

$$\Delta = (\text{test error at final exit}) - (\text{test error if exited at correct prediction})$$

$$\Delta(\text{CIFAR-100}) = \sim 14\%$$

$$\Delta(\text{ImageNet}) = \sim 9\%$$

Early-Exit Neural Networks

⊗ interruptibility



⊗ monotonicity



⊗ diminishing returns?

Early-Exit Neural Networks

- ⊗ interruptibility

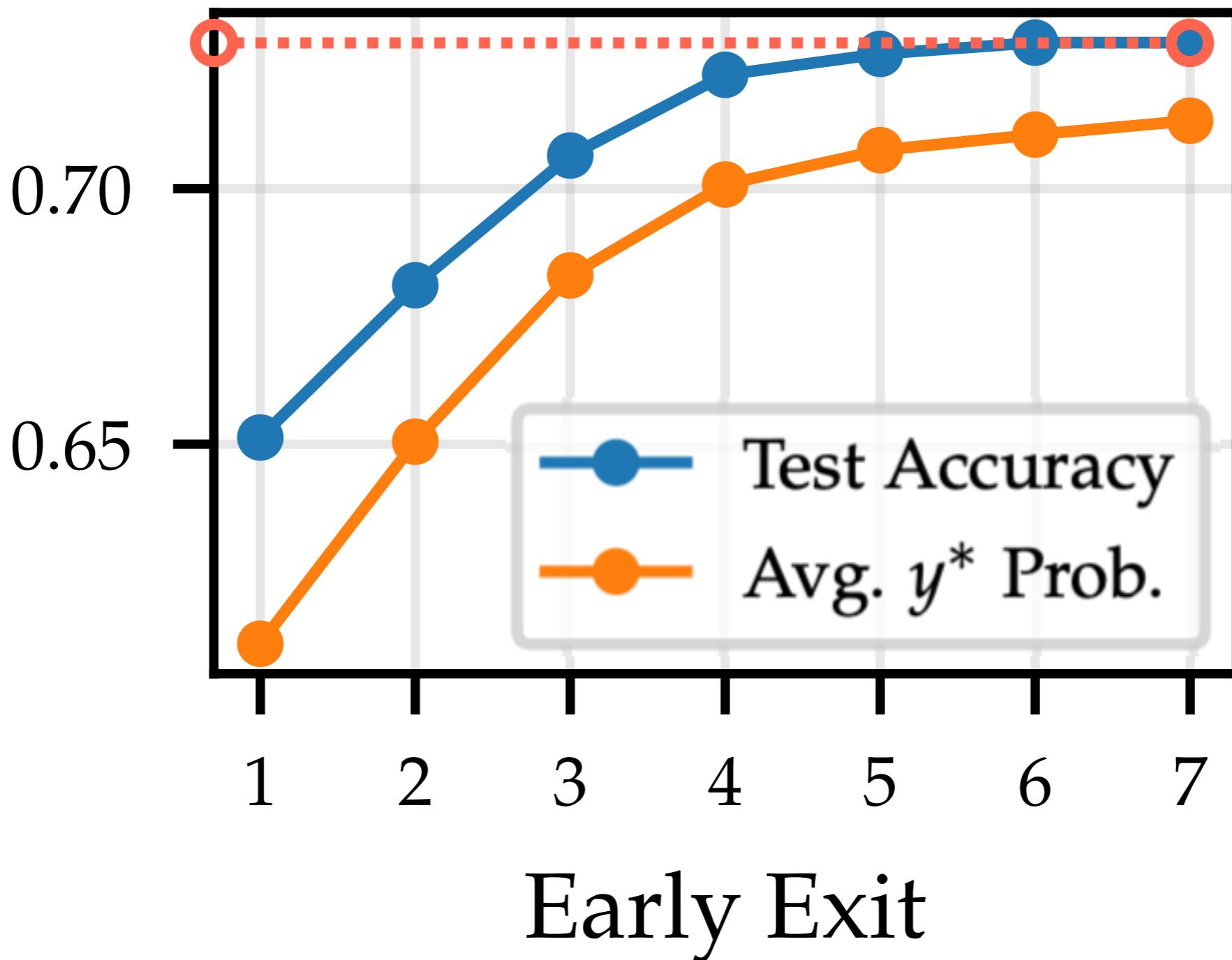


- ⊗ monotonicity



- ⊗ diminishing returns?

Multi-Scale Dense Net: CIFAR-100



Early-Exit Neural Networks

- ⊗ interruptibility



- ⊗ monotonicity



- ⊗ diminishing returns



Early-Exit Neural Networks

⊗ interruptibility



⊗ monotonicity



⊗ diminishing returns



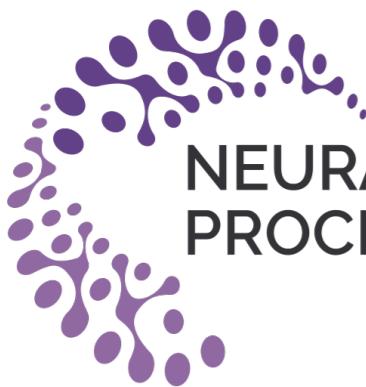
only marginally

Early-Exit Neural Networks

- ⊗ interruptibility
- ⊗ monotonicity
- ⊗ diminishing returns



A simple, post-hoc method for encouraging conditional monotonicity



NEURAL INFORMATION
PROCESSING SYSTEMS

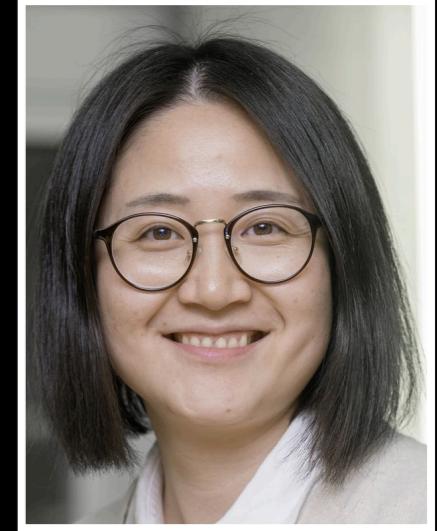
2023



Metod
Jazbec

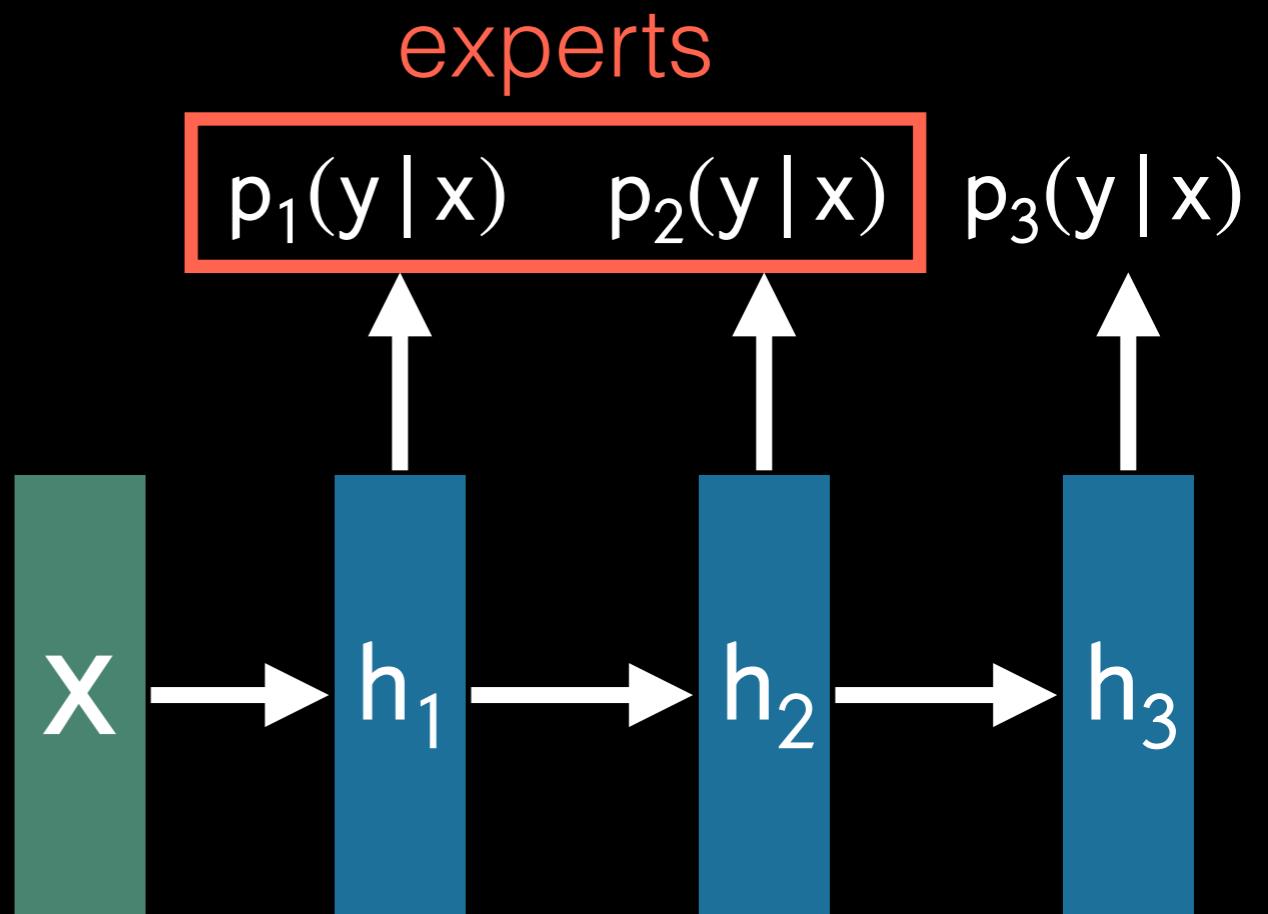


James U.
Allingham

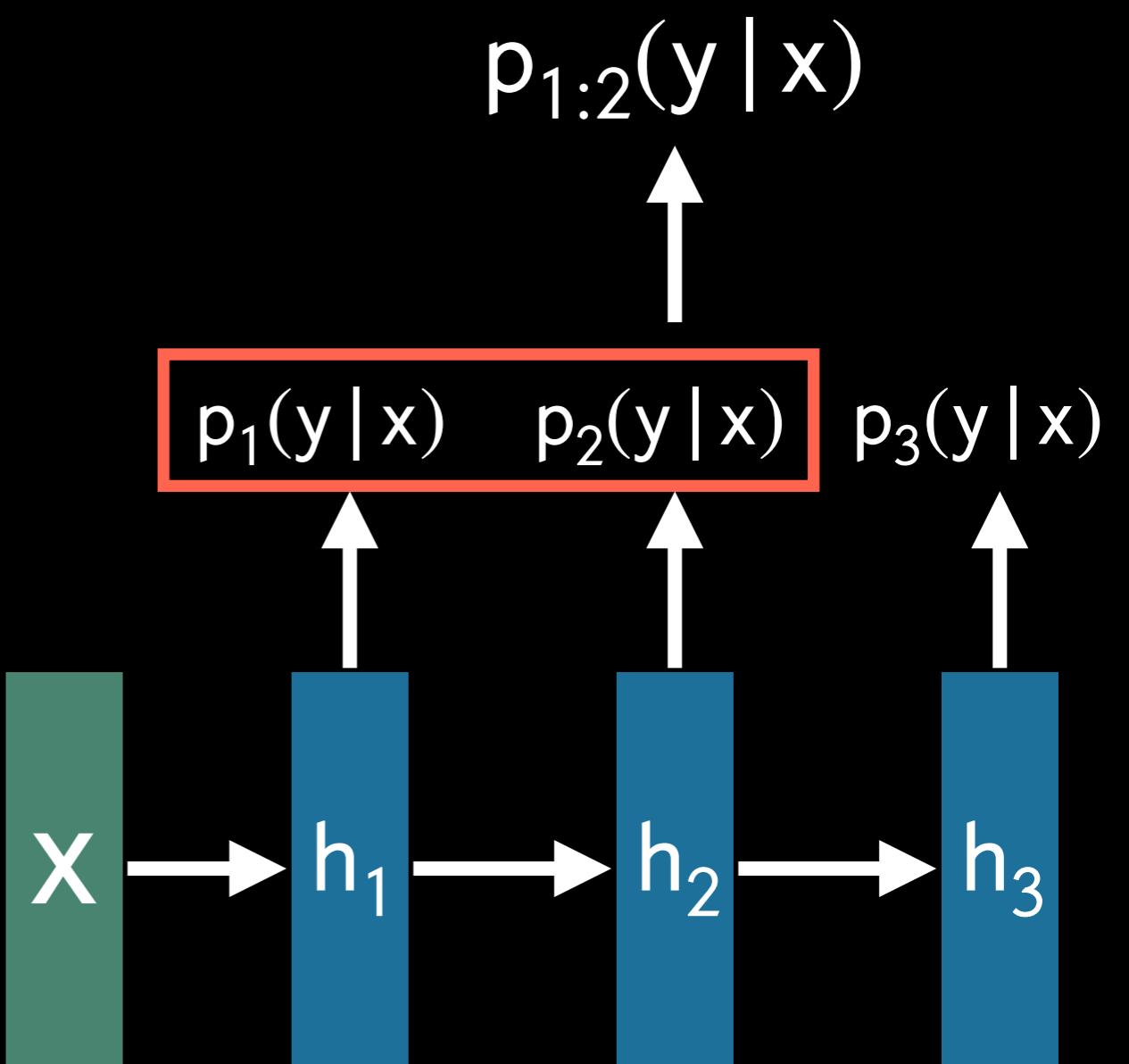


Dan
Zhang

Idea: combine the early-exits
via a product of experts



Idea: combine the early-exits
via a product of experts



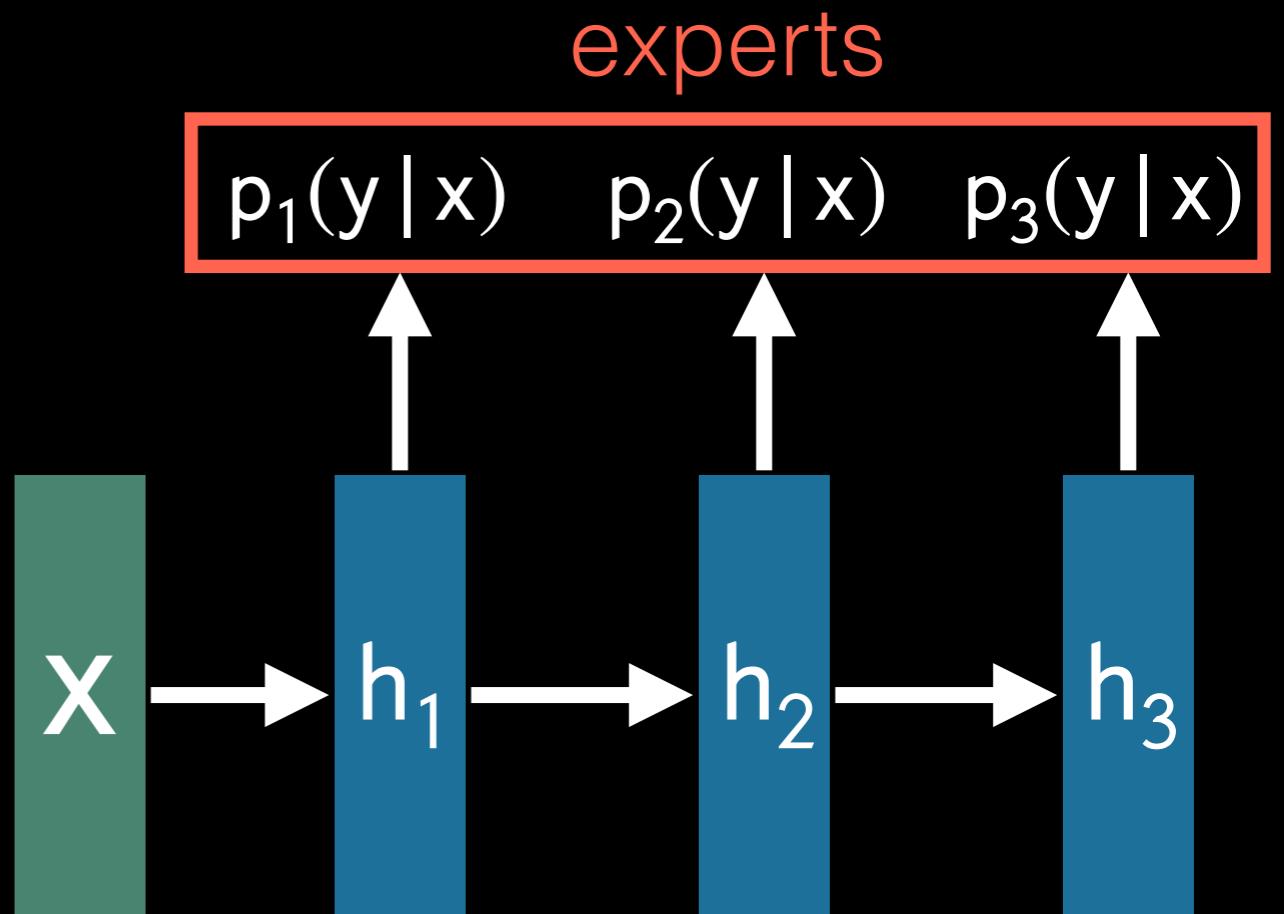
Idea: combine the early-exits
via a product of experts

$$p_{1:2}(y|x) = \frac{p_1(y|x) \cdot p_2(y|x)}{\sum_{y'} p_1(y'|x) \cdot p_2(y'|x)}$$

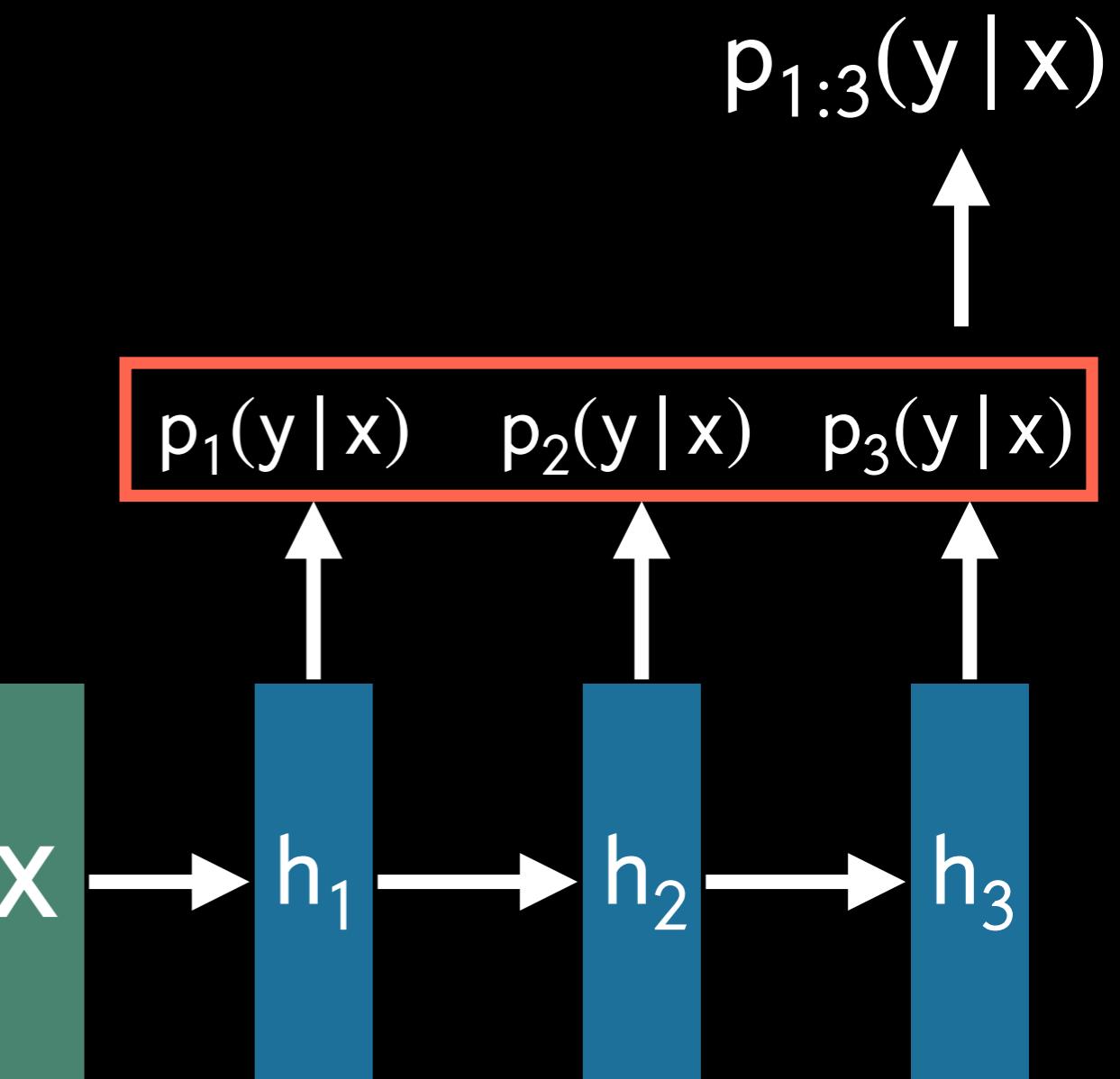
$p_{1:2}(y|x)$

The diagram illustrates a neural network architecture. It starts with an input layer labeled x , represented by a green rectangle. An arrow points from x to the first hidden layer, labeled h_1 , represented by a blue rectangle. Another arrow points from h_1 to the second hidden layer, labeled h_2 , also represented by a blue rectangle. A third arrow points from h_2 to the final hidden layer, labeled h_3 , also represented by a blue rectangle. Above the network, a red box highlights the outputs of the first two layers: $p_1(y|x)$ and $p_2(y|x)$. An upward arrow points from this red box to the equation for $p_{1:2}(y|x)$. To the right of the red box, another upward arrow points to the equation for $p_{1:2}(y|x)$.

Idea: combine the early-exits
via a product of experts



Idea: combine the early-exits
via a product of experts



Idea: combine the early-exits
via a product of experts

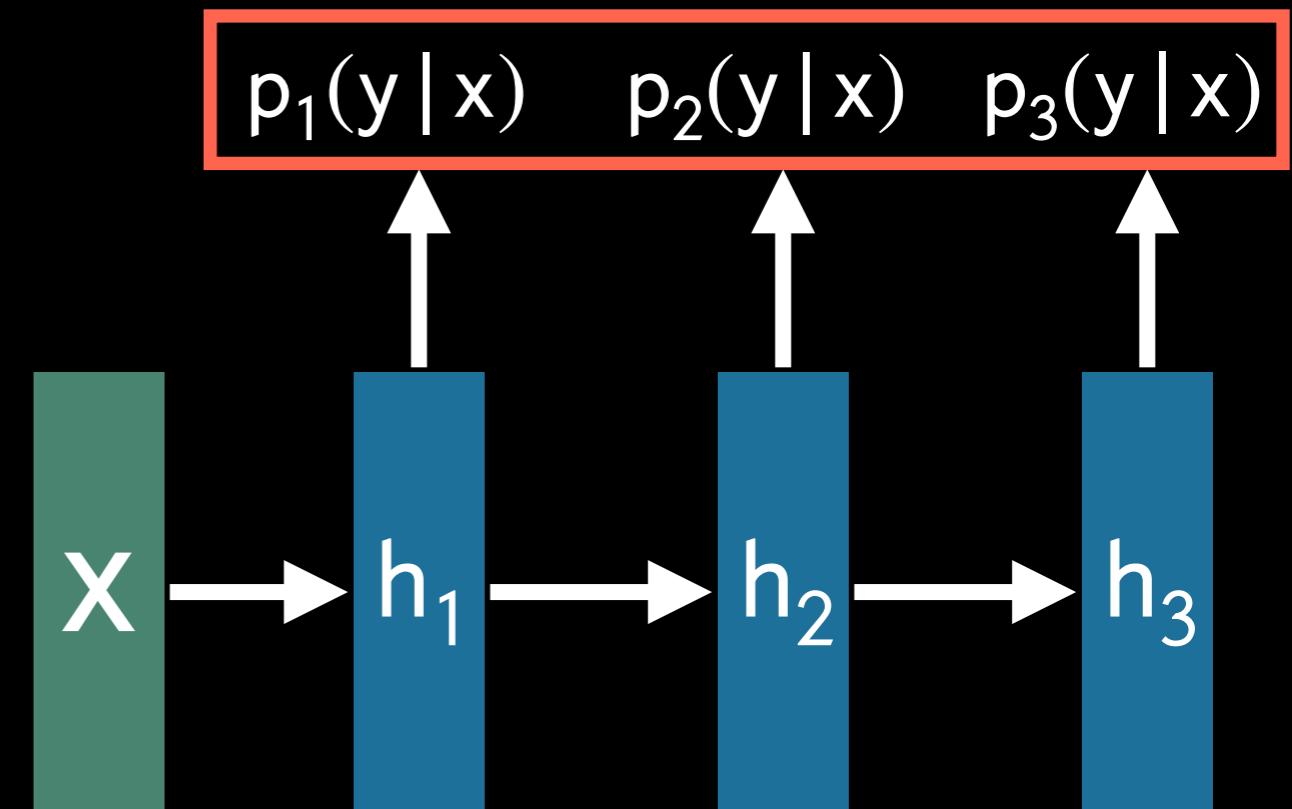
$$p_{1:3}(y|x) = \frac{p_1(y|x) \cdot p_2(y|x) \cdot p_3(y|x)}{\sum_{y'} p_1(y'|x) \cdot p_2(y'|x) \cdot p_3(y'|x)}$$

$p_{1:3}(y|x)$

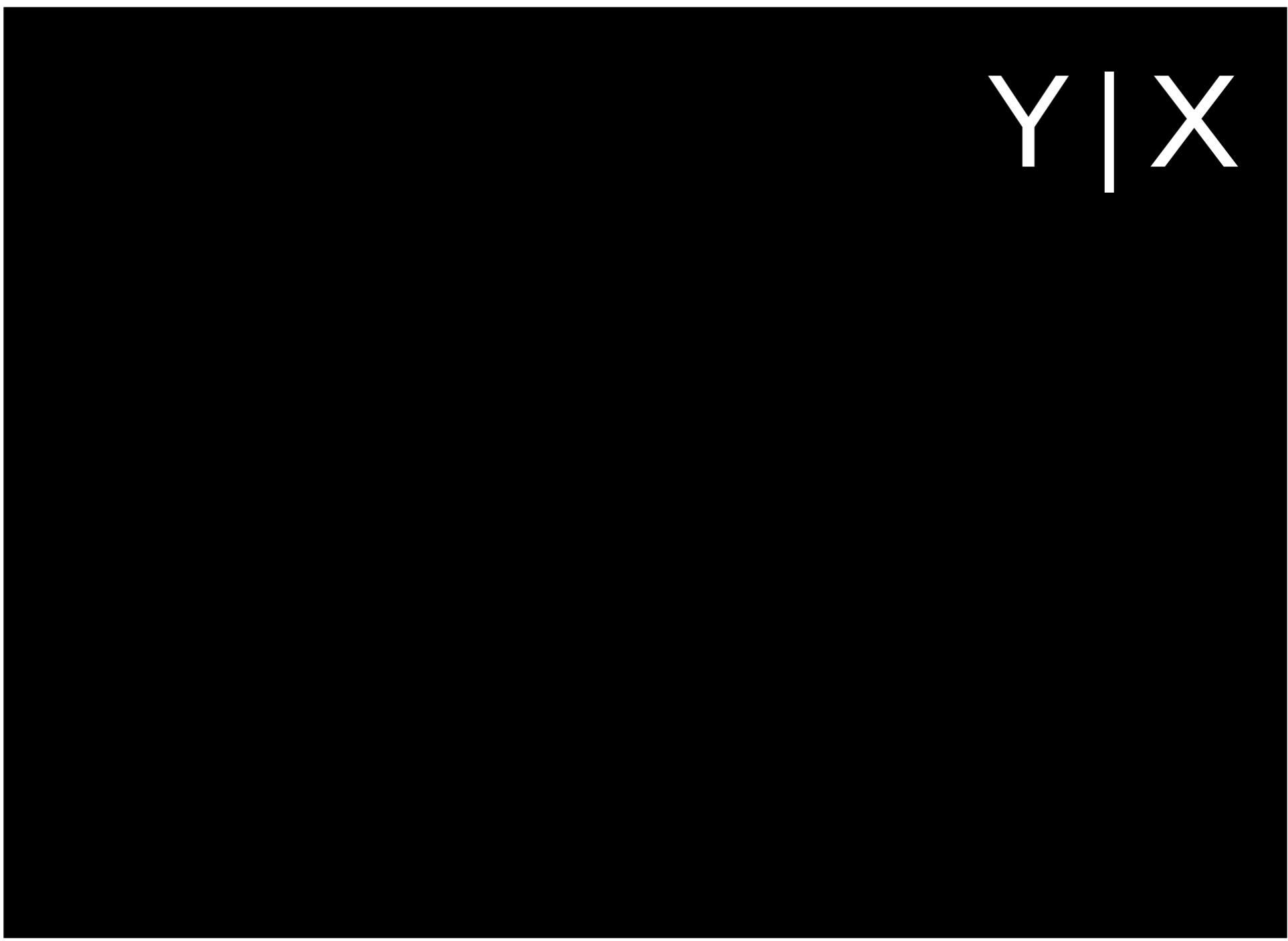
The diagram illustrates a neural network architecture for combining expert predictions. It starts with an input layer labeled x , followed by three hidden layers labeled h_1 , h_2 , and h_3 . Arrows indicate the flow of information from x through h_1 , h_2 , and h_3 . Above the network, the formula for the combined probability $p_{1:3}(y|x)$ is shown, where the terms $p_1(y|x)$, $p_2(y|x)$, and $p_3(y|x)$ are highlighted with a red border. Arrows point from each h_i layer to its corresponding term in the formula.

Idea: combine the early-exits
via a product of experts

$$p_{1:e}(y|x) = \frac{\prod_{j=1}^e p_j(y_j|x)}{\sum_{y'} \prod_{j=1}^e p_j(y'_j|x)}$$



Idea: combine the early-exits
via a product of experts



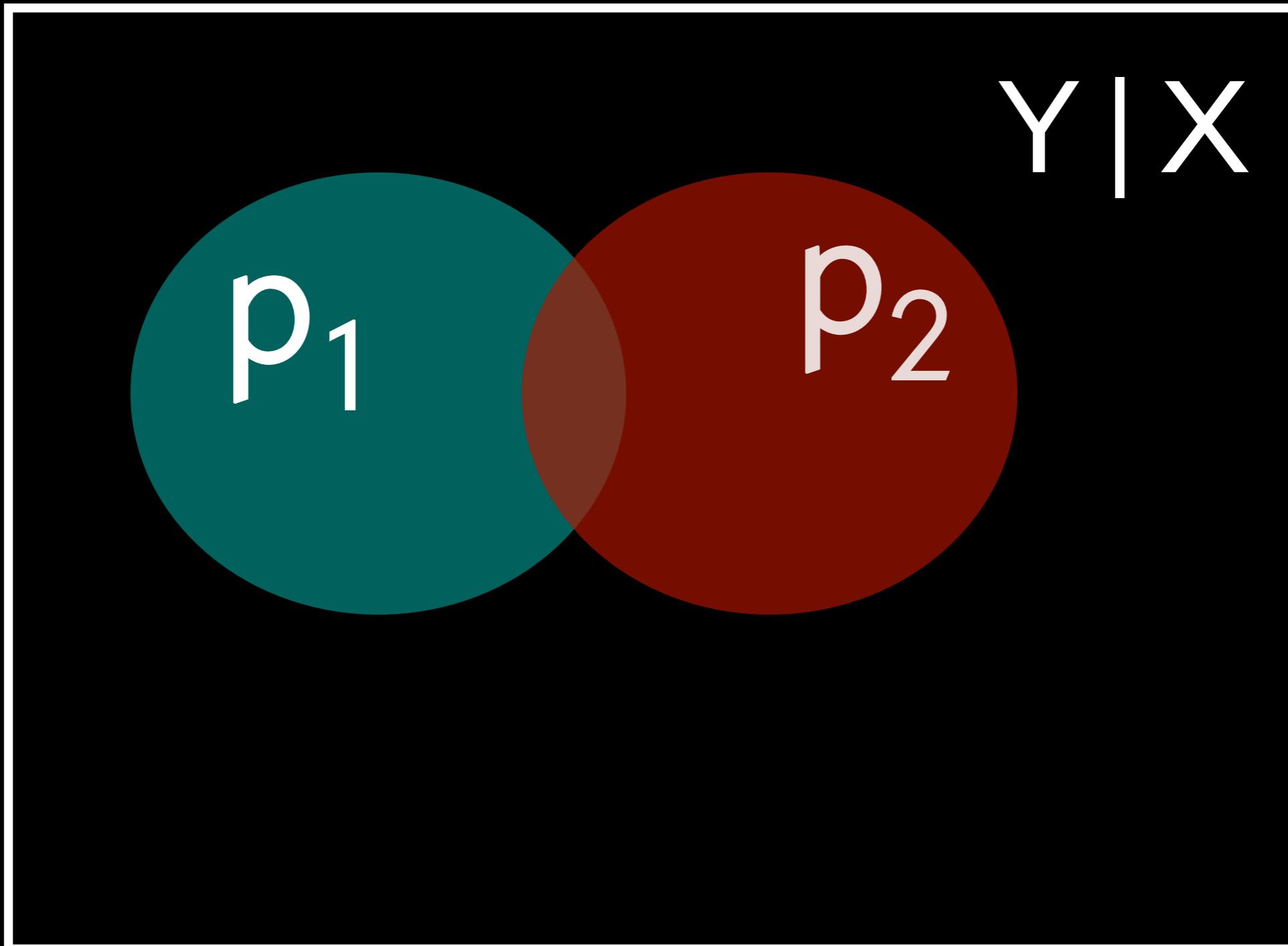
Y | X

Idea: combine the early-exits
via a product of experts

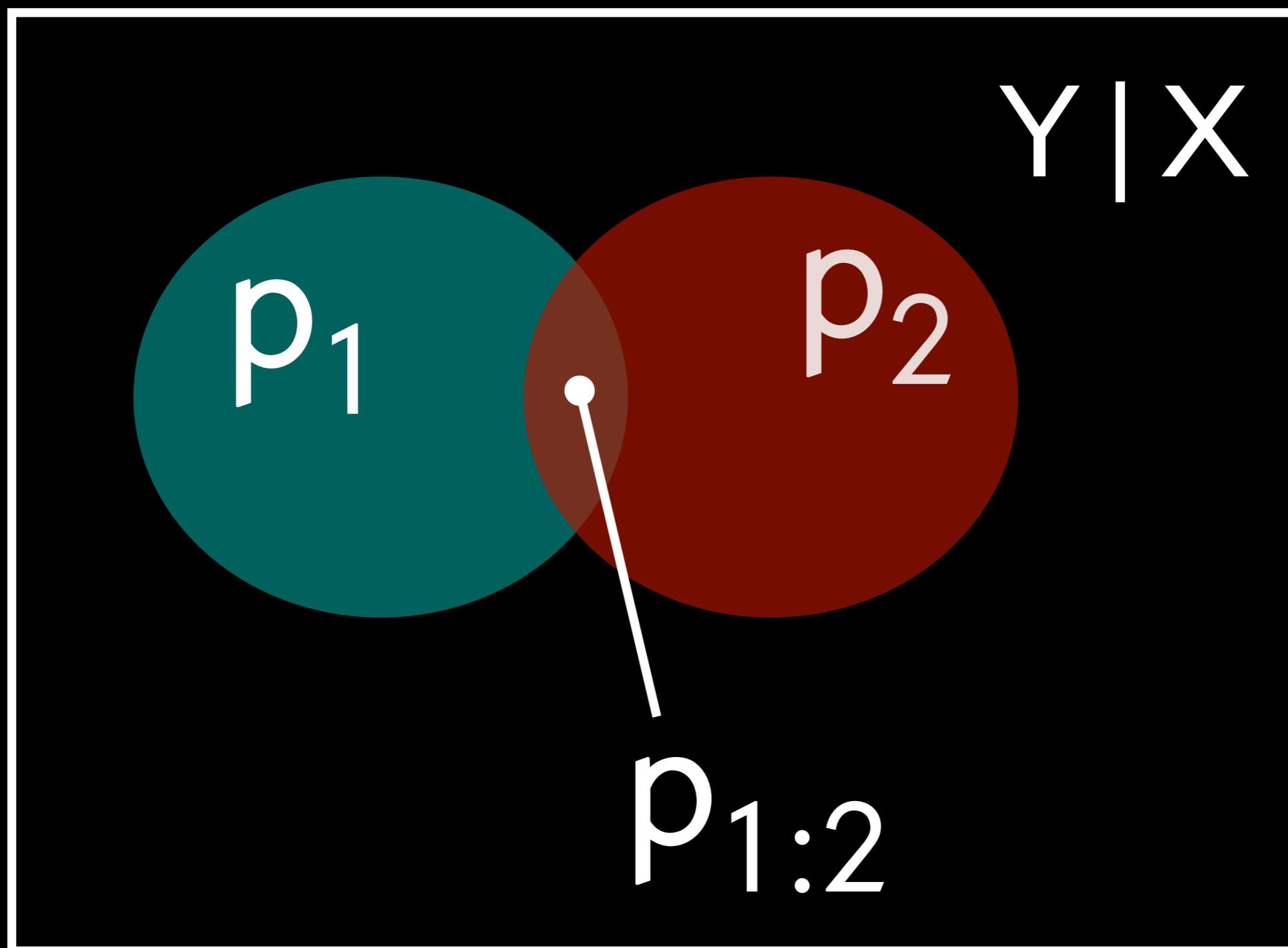
$$Y | X$$

p_1

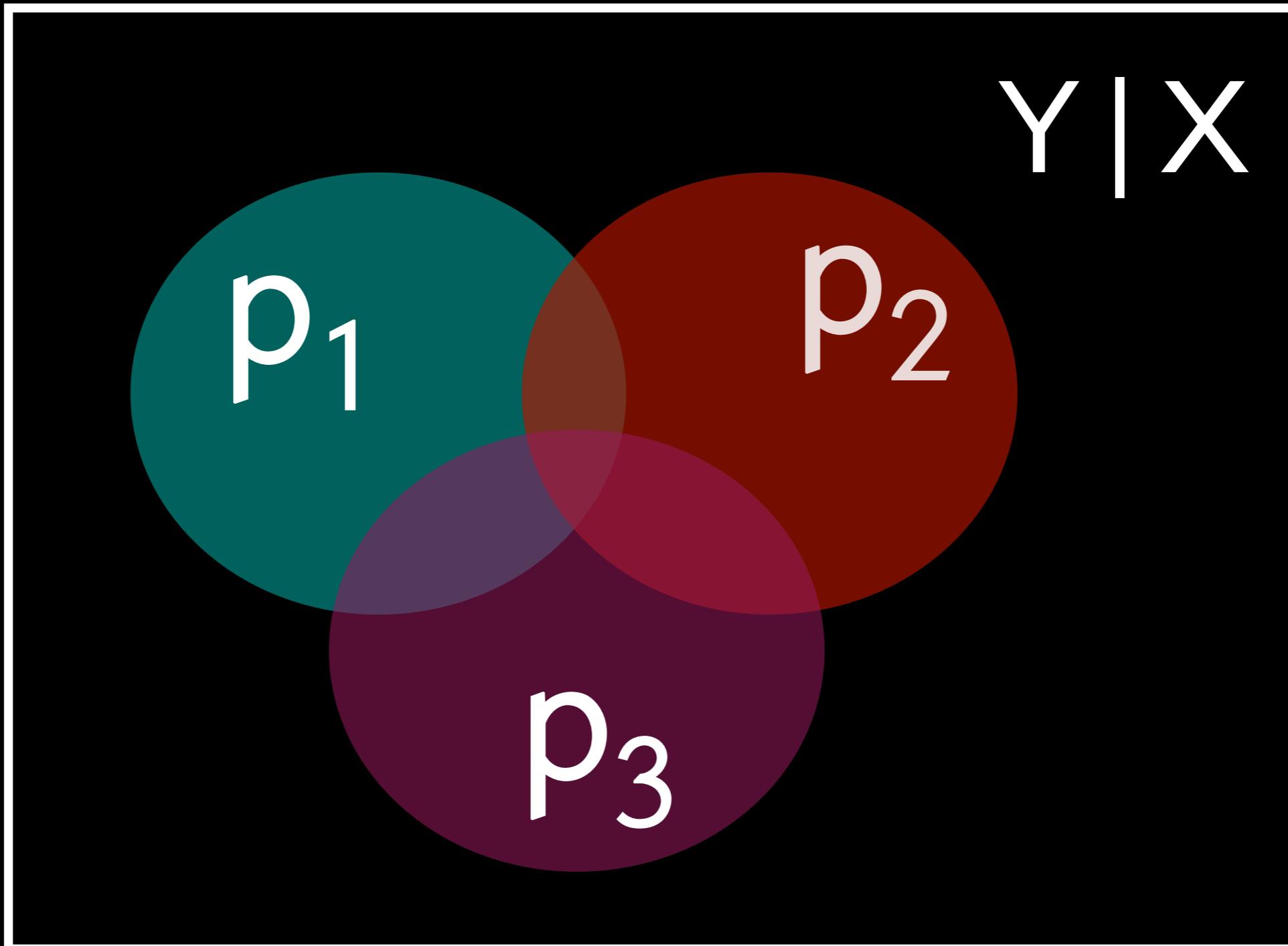
Idea: combine the early-exits
via a product of experts



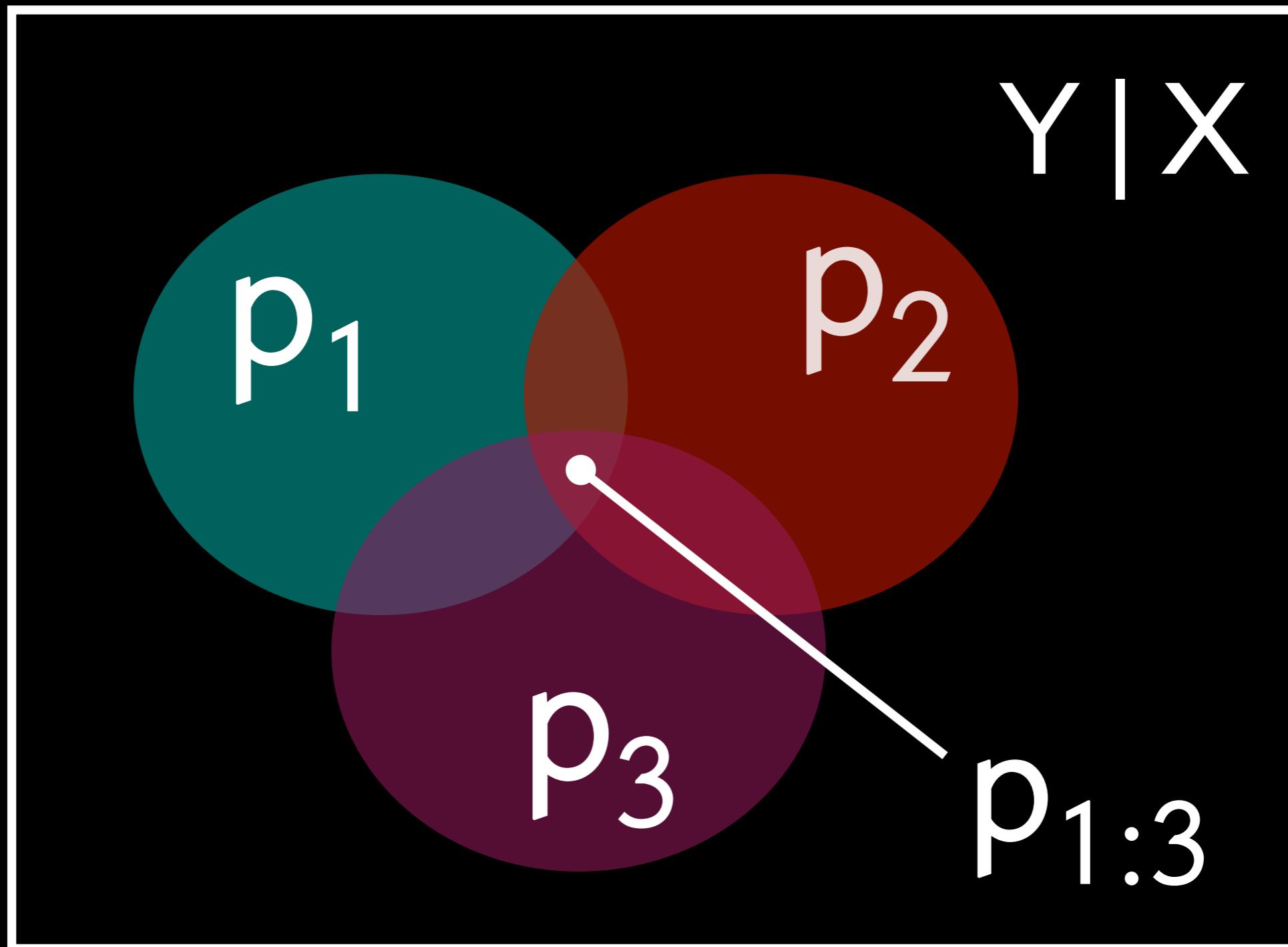
Idea: combine the early-exits
via a product of experts



Idea: combine the early-exits
via a product of experts



Idea: combine the early-exits
via a product of experts



Idea: combine the early-exits
via a product of experts

One catch: exit distributions must have finite
(or quickly decaying) support to bound
influence of $(e+1)$ th expert.

Idea: combine the early-exits
via a product of experts

One catch: exit distributions must have finite
(or quickly decaying) support to bound
influence of $(e+1)$ th expert.

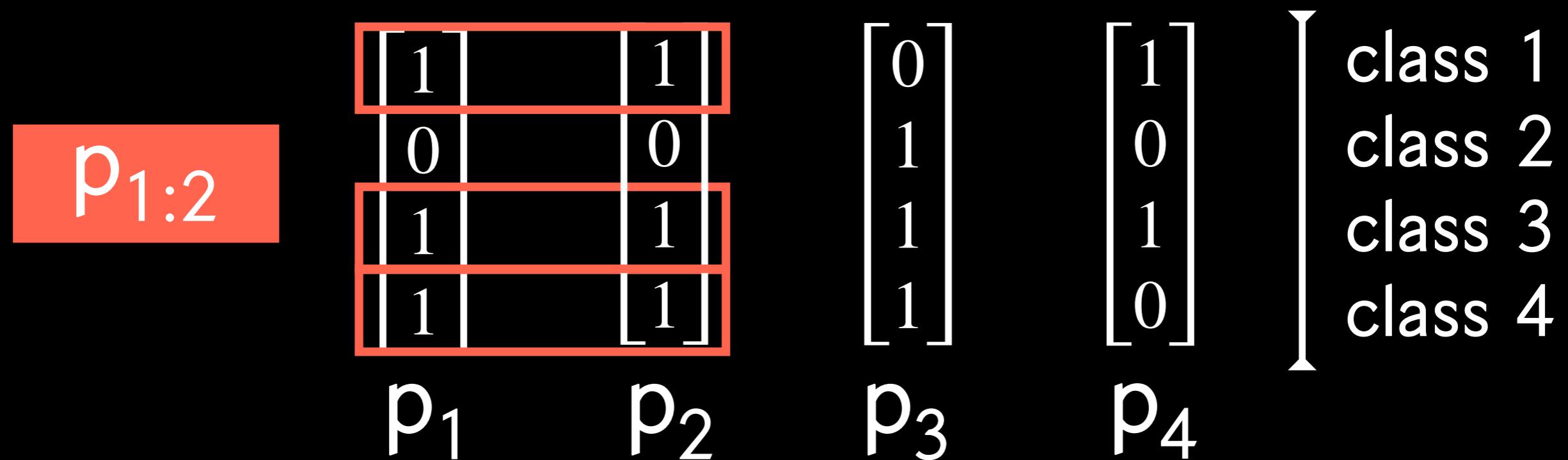
Ideal case: binary one-vs-rest.

$$\begin{array}{c} \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} \\ p_1 \quad p_2 \quad p_3 \quad p_4 \end{array} \quad \left. \right| \begin{array}{l} \text{class 1} \\ \text{class 2} \\ \text{class 3} \\ \text{class 4} \end{array}$$

Idea: combine the early-exits
via a product of experts

One catch: exit distributions must have finite
(or quickly decaying) support to bound
influence of $(e+1)$ th expert.

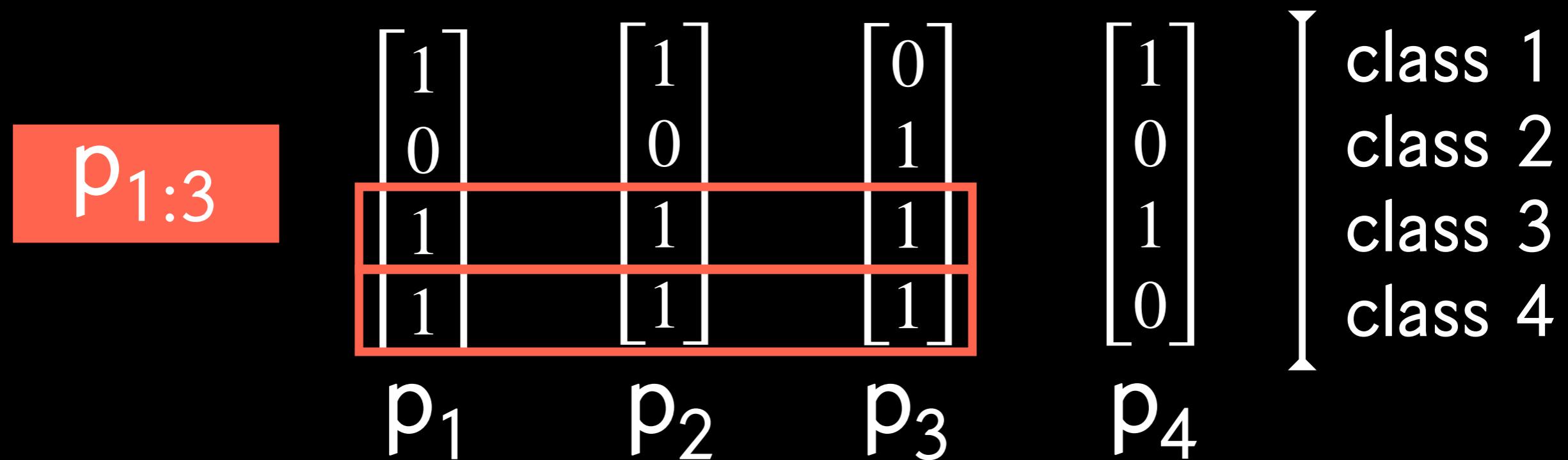
Ideal case: binary one-vs-rest.



Idea: combine the early-exits
via a product of experts

One catch: exit distributions must have finite
(or quickly decaying) support to bound
influence of $(e+1)$ th expert.

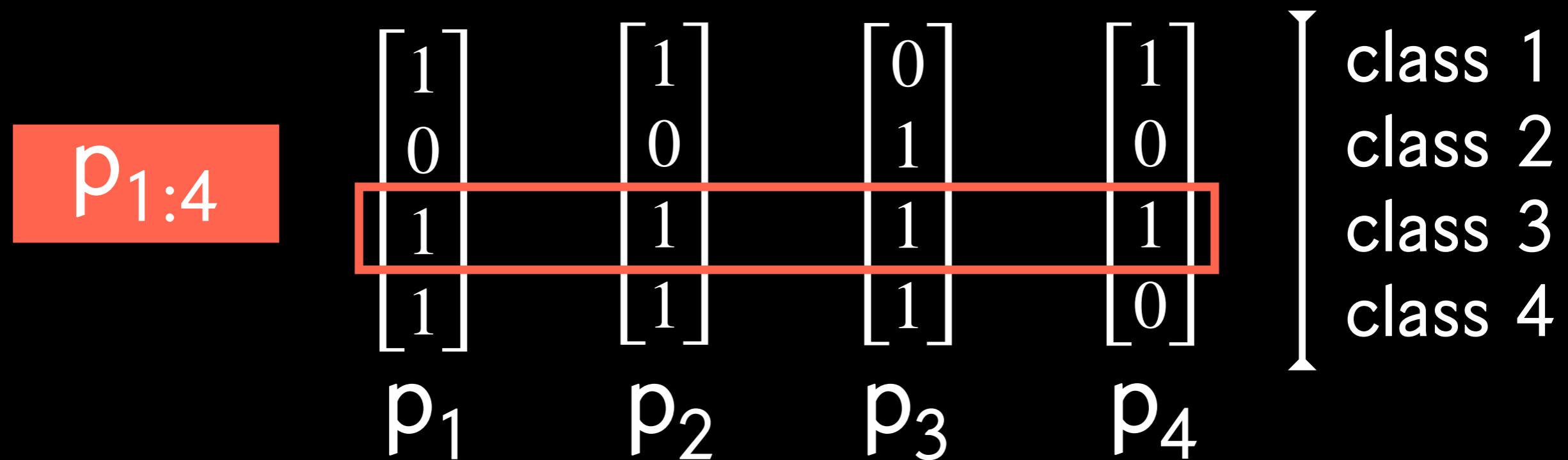
Ideal case: binary one-vs-rest.



Idea: combine the early-exits
via a product of experts

One catch: exit distributions must have finite
(or quickly decaying) support to bound
influence of $(e+1)$ th expert.

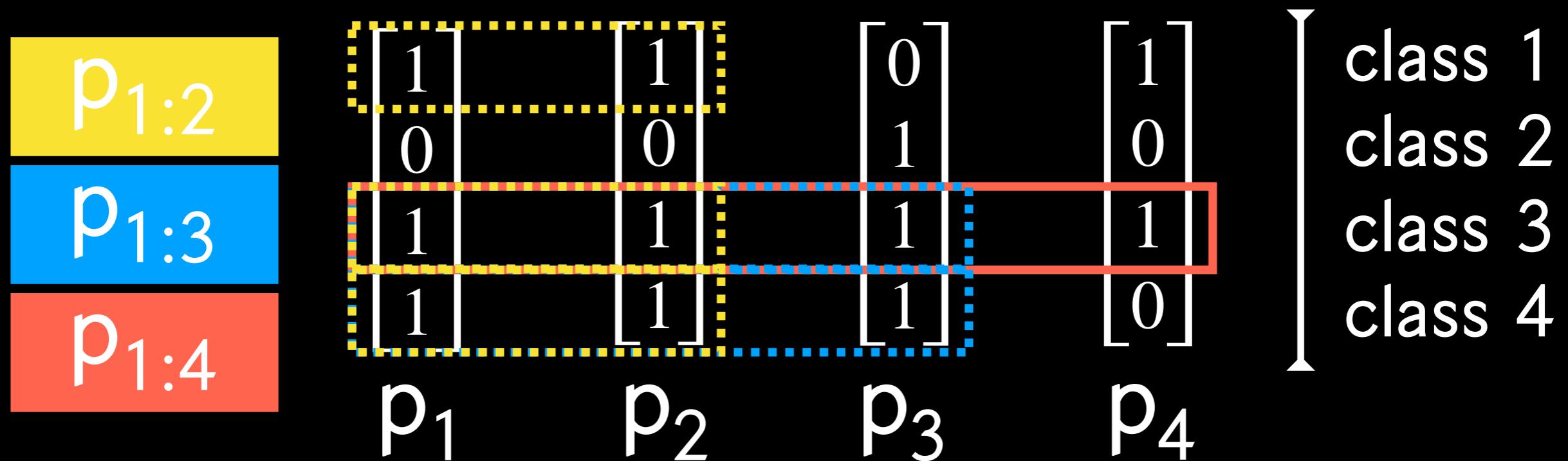
Ideal case: binary one-vs-rest.



Idea: combine the early-exits
via a product of experts

One catch: exit distributions must have finite
(or quickly decaying) support to bound
influence of $(e+1)$ th expert.

Ideal case: binary one-vs-rest.



Implementation with ReLUs

$$p_{1:e}(y | x) = \frac{\prod_{j=1}^e \max(0, f_{j,y}(x))}{\sum_{y'} \prod_{j=1}^e \max(0, f_{j,y'}(x))}$$

$f_{j,y}(x)$ is logit for y th class at j th exit

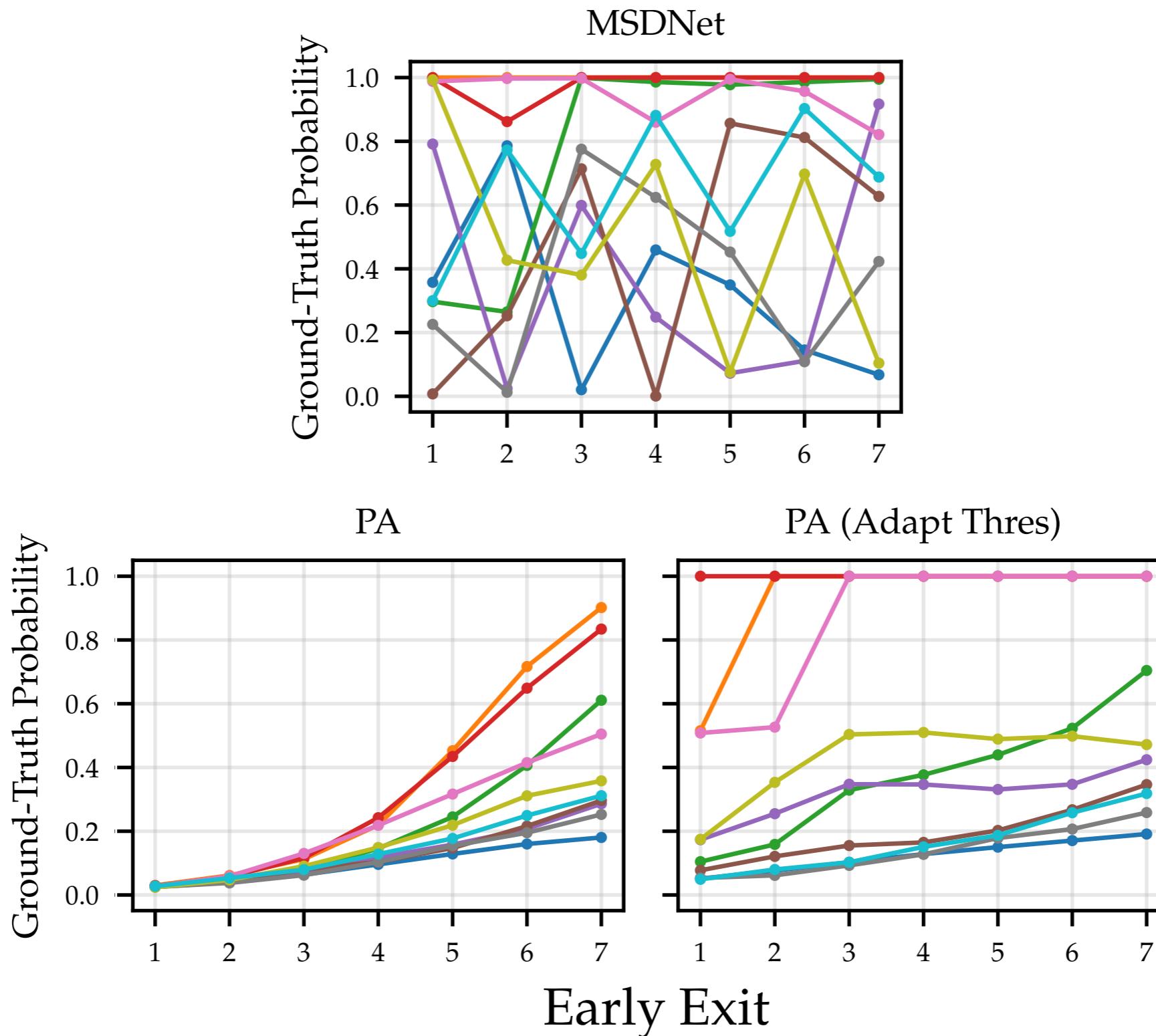
Implementation with ReLUs

$$p_{1:e}(y | x) = \frac{\prod_{j=1}^e \max(0, f_{j,y}(x))}{\sum_{y'} \prod_{j=1}^e \max(0, f_{j,y'}(x))}$$

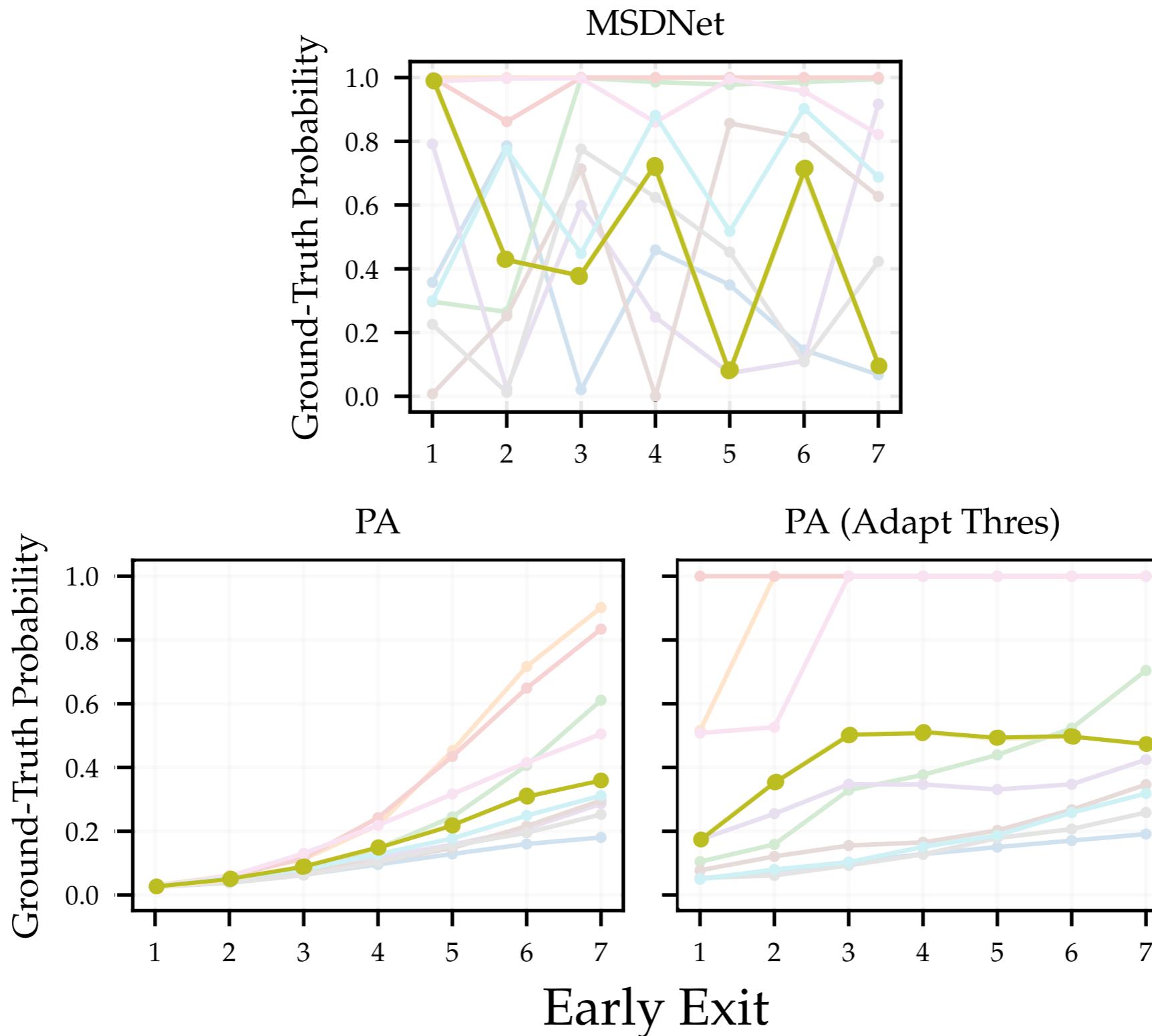
Clipping logits controls deviation from perfect monotonicity.

We apply this transformation post-hoc!

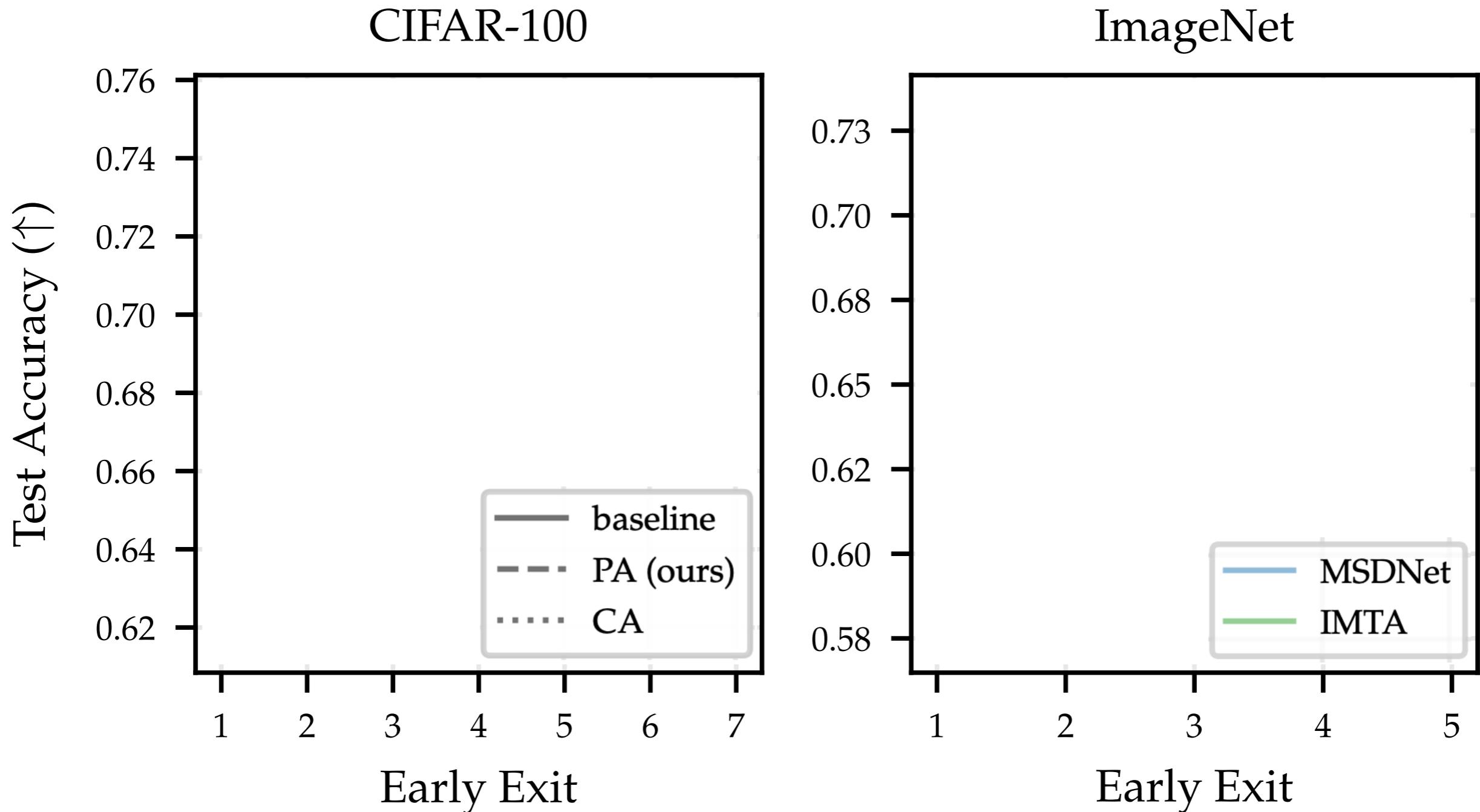
Monotonicity: CIFAR-100



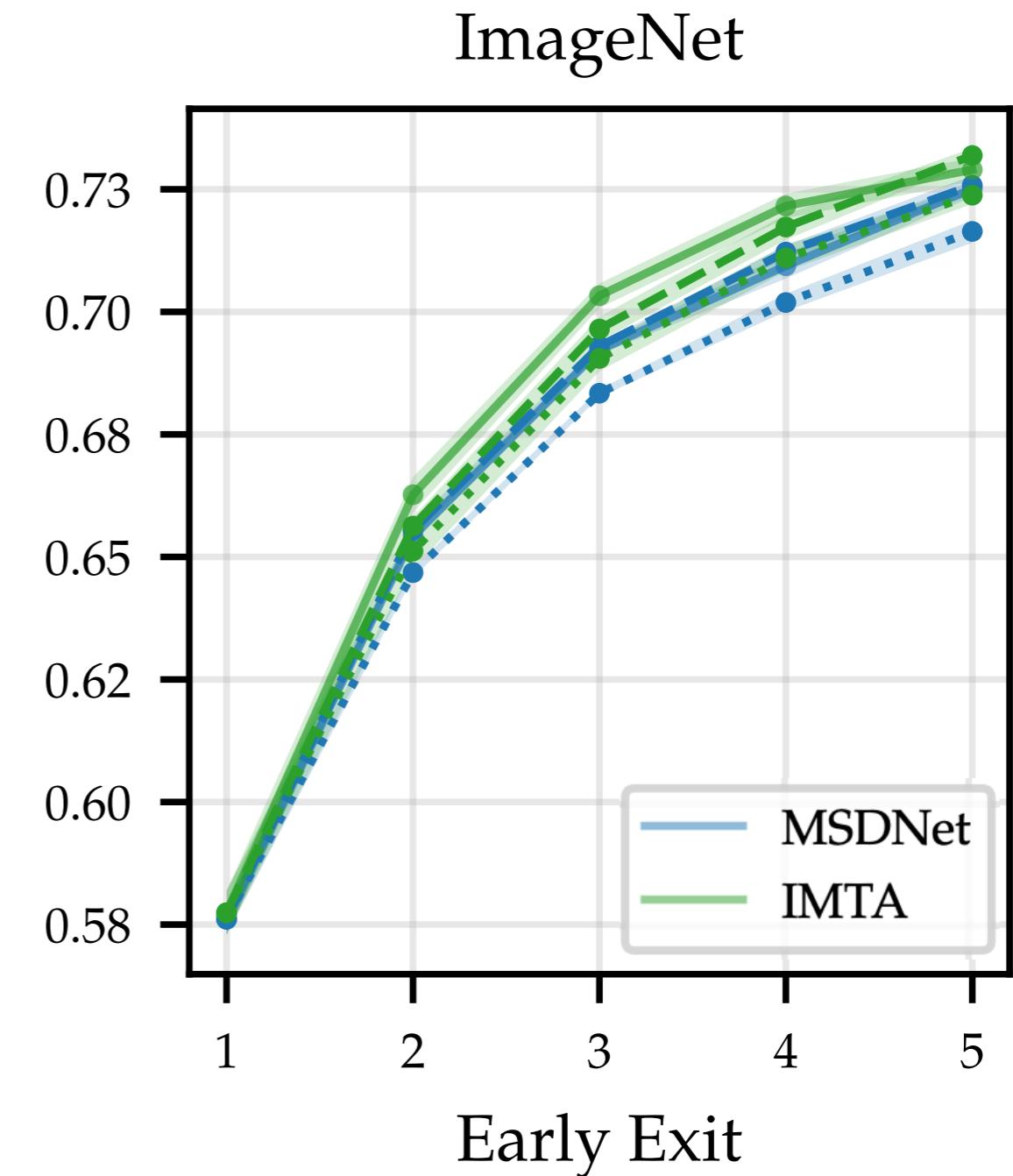
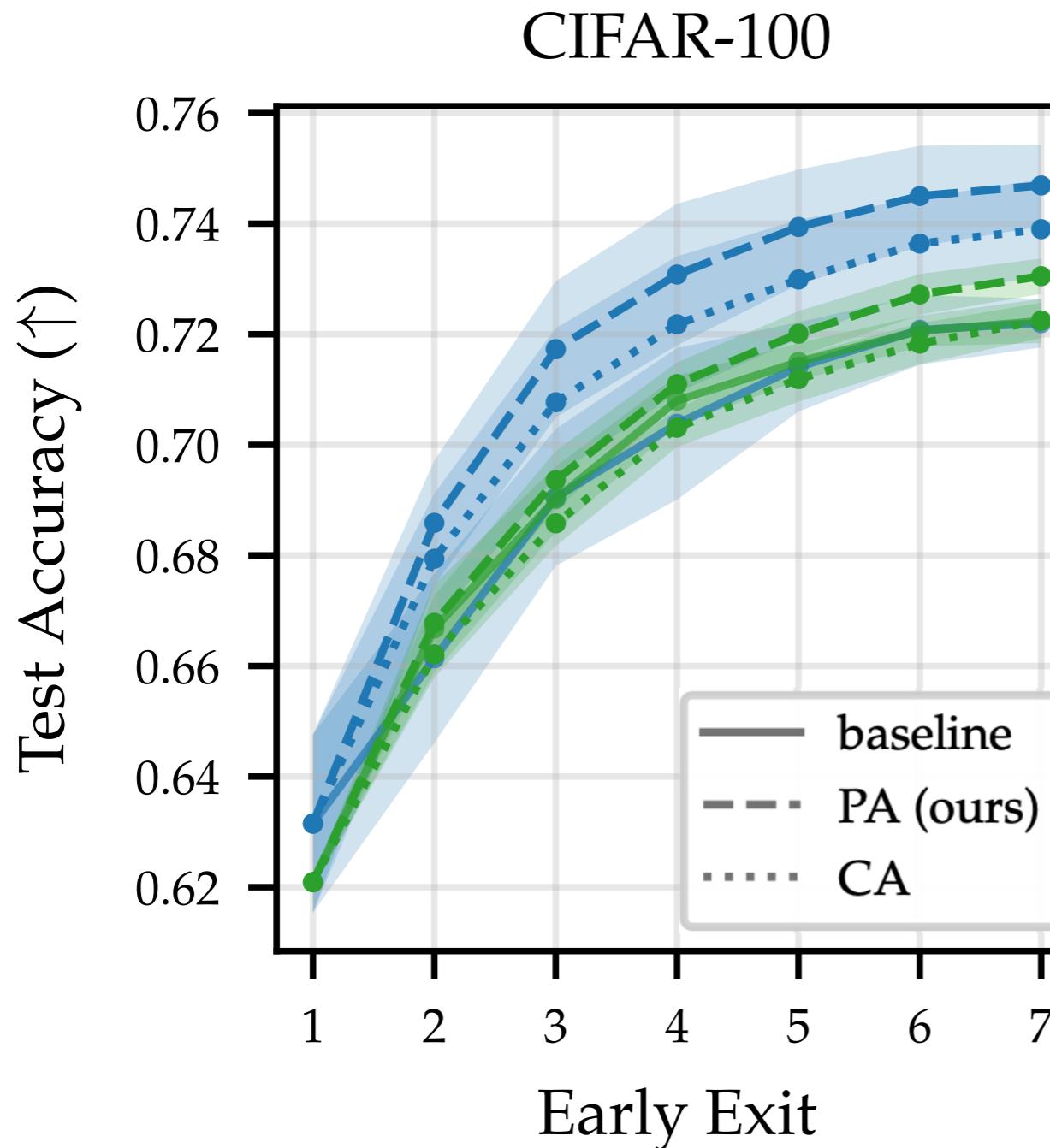
Monotonicity: CIFAR-100



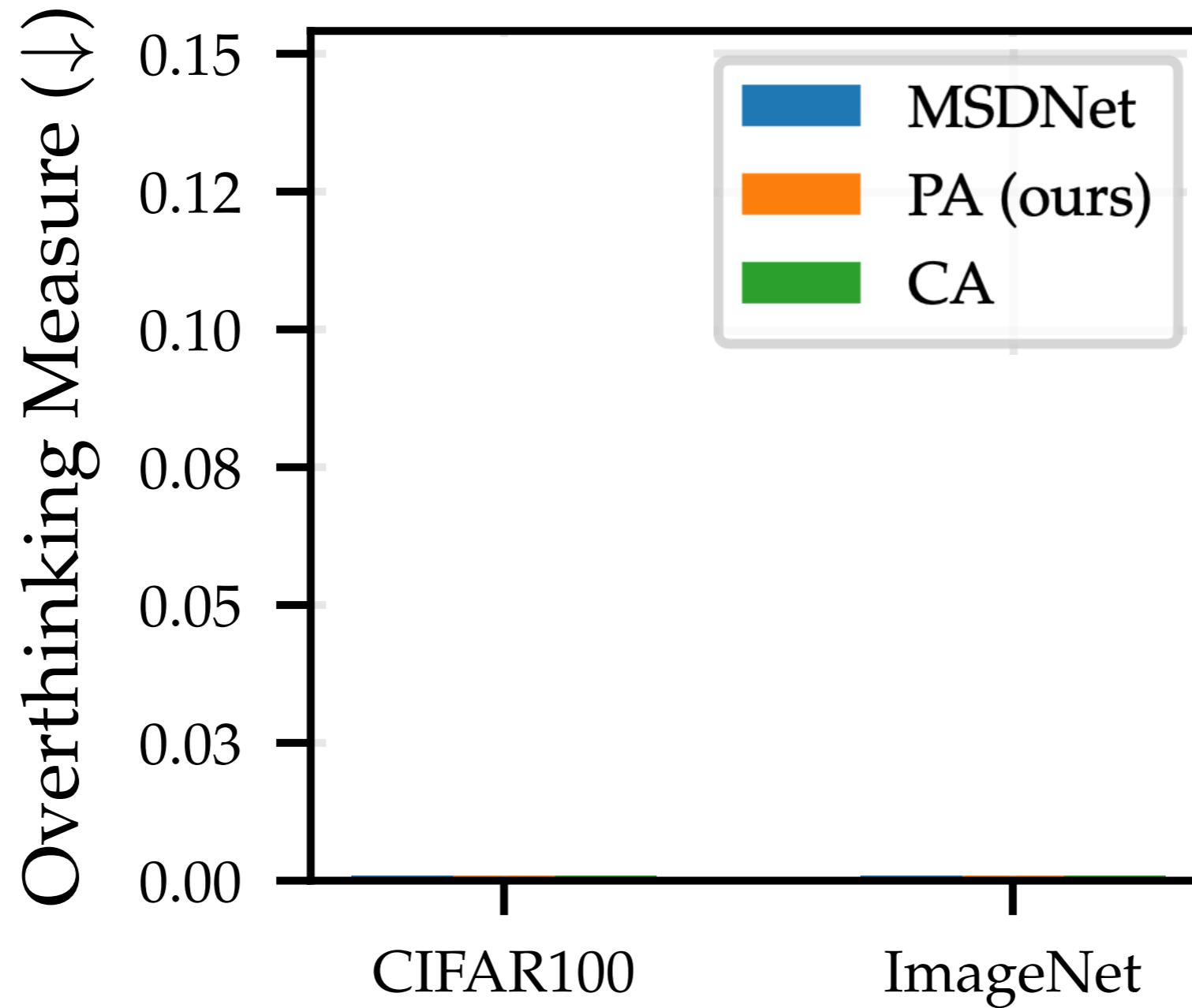
Accuracy: CIFAR-100 & ImageNet



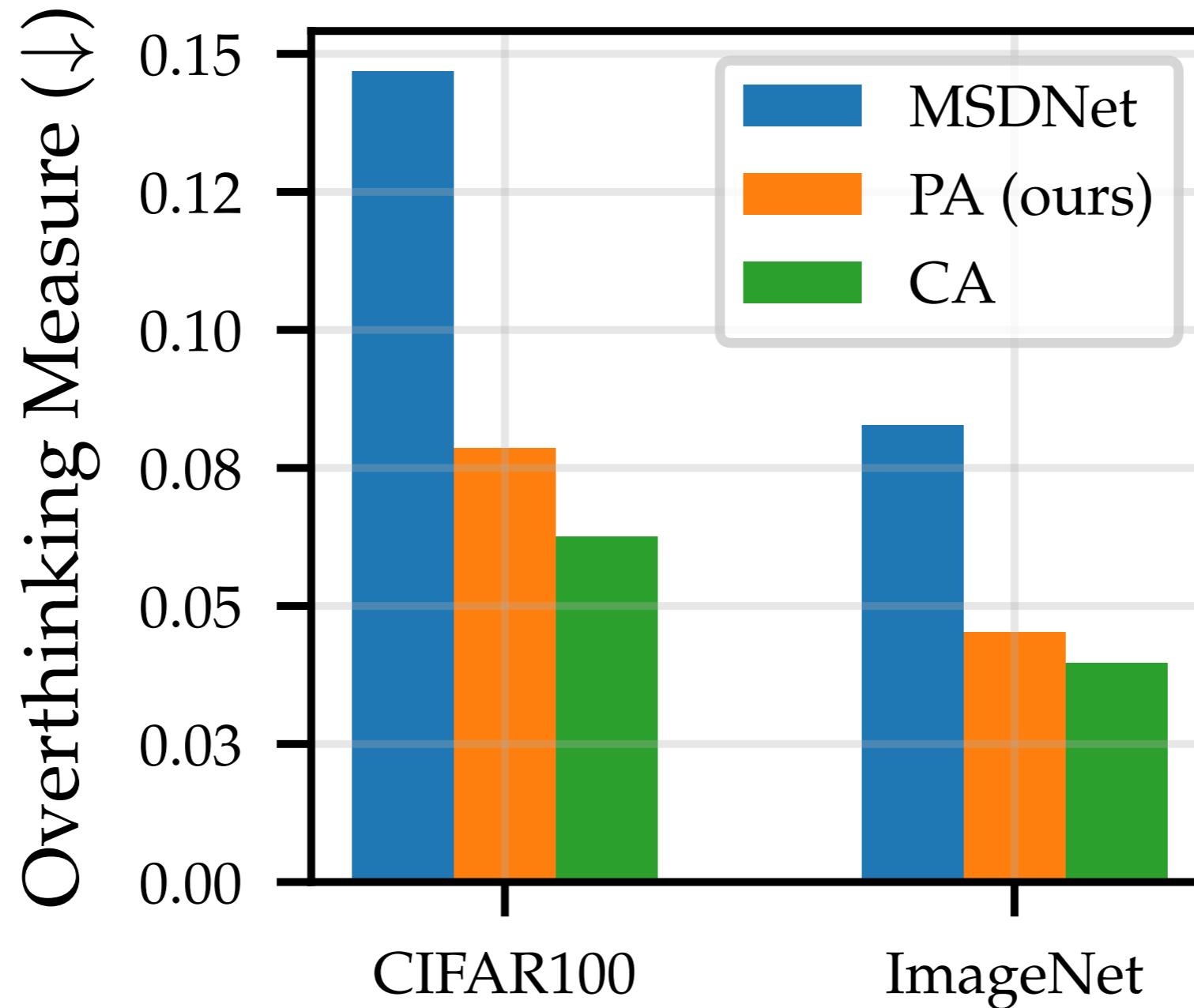
Accuracy: CIFAR-100 & ImageNet



Overthinking: CIFAR-100 & ImageNet

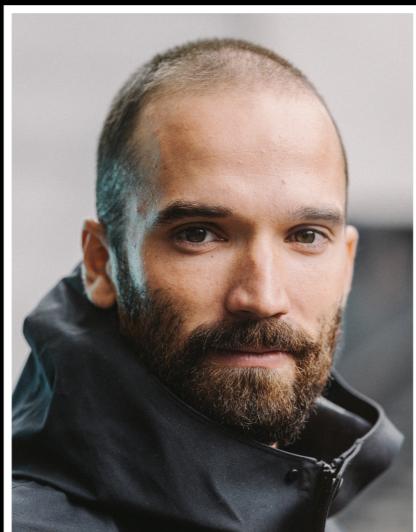


Overthinking: CIFAR-100 & ImageNet



* Doesn't mean that overall accuracy is improved by this amount since our model makes more mistakes at intermediate exits.

Ensuring consistency across exits in predictive uncertainty estimates



Metod
Jazbec



Dan
Zhang

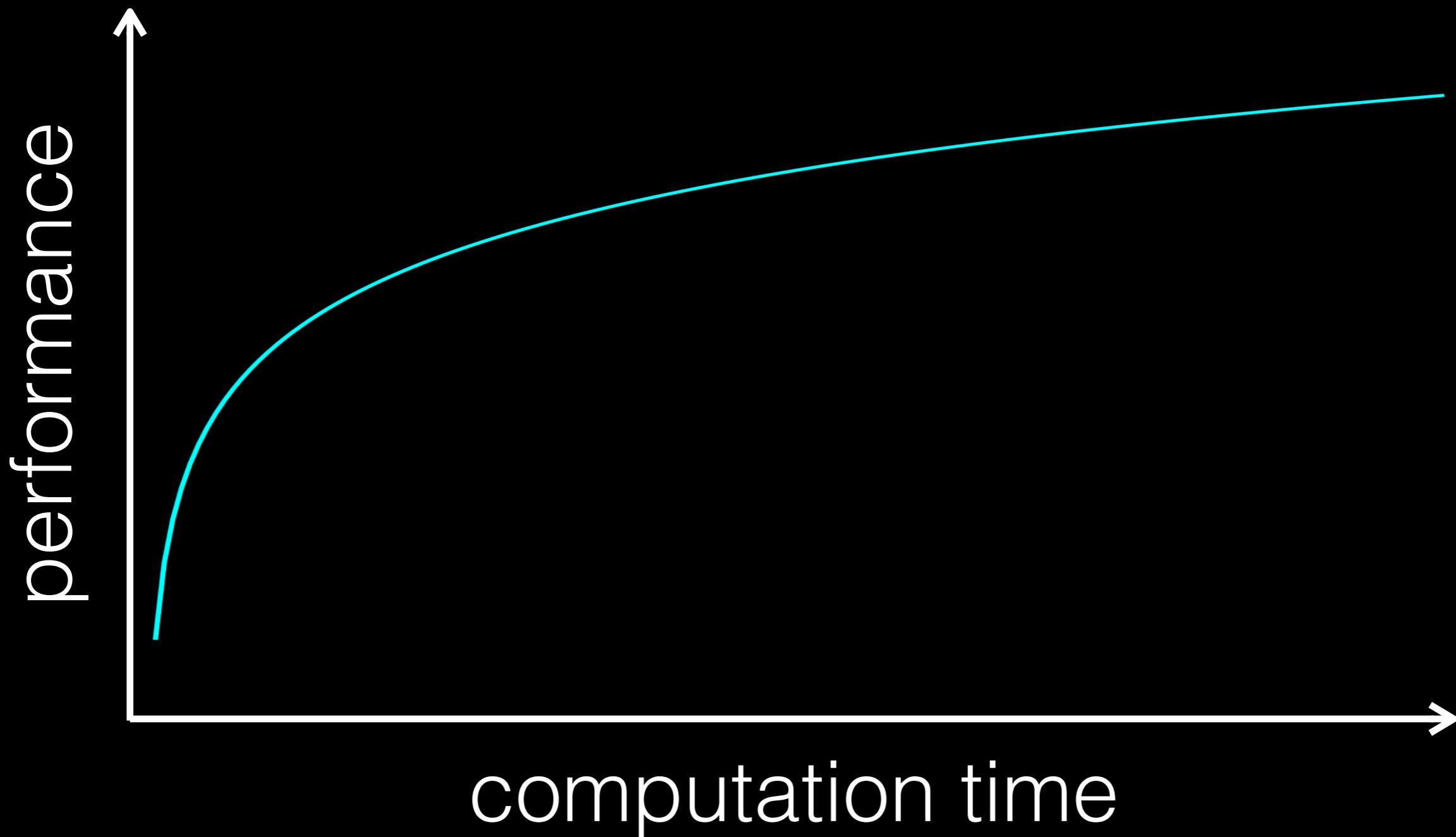


Patrick
Forré

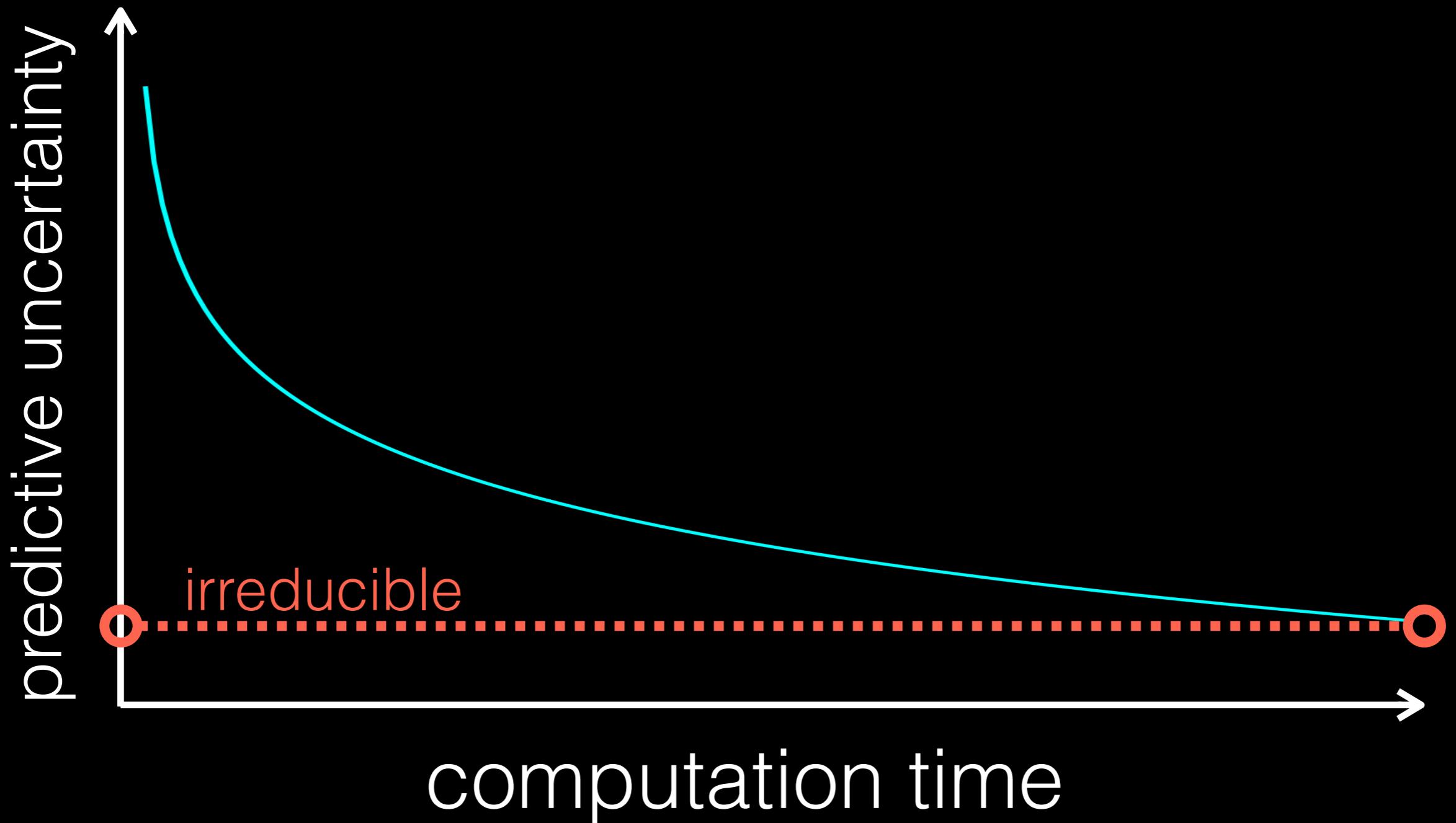


Stephan
Mandt

Anytime Models



Anytime Uncertainty

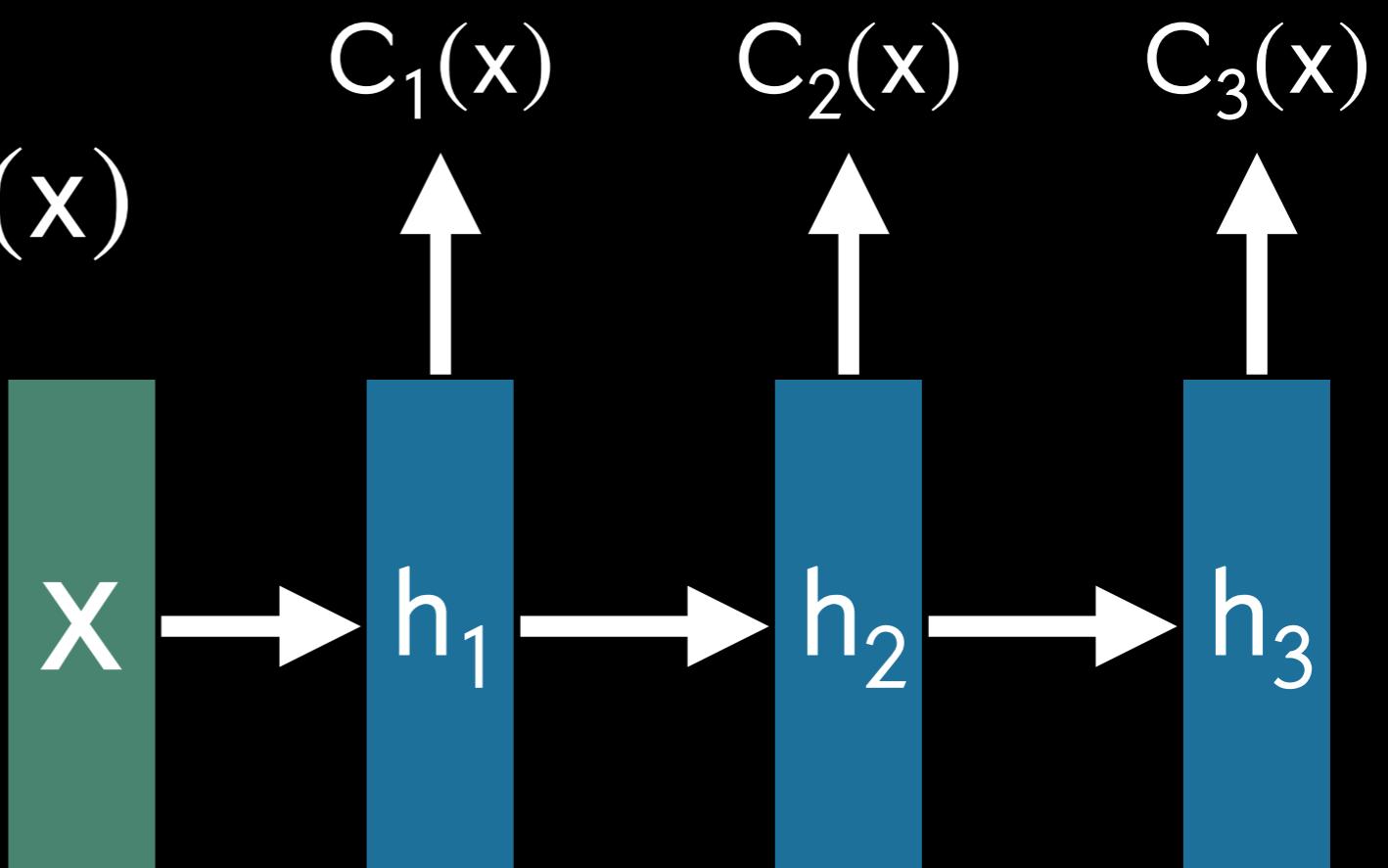


Anytime Uncertainty Estimation

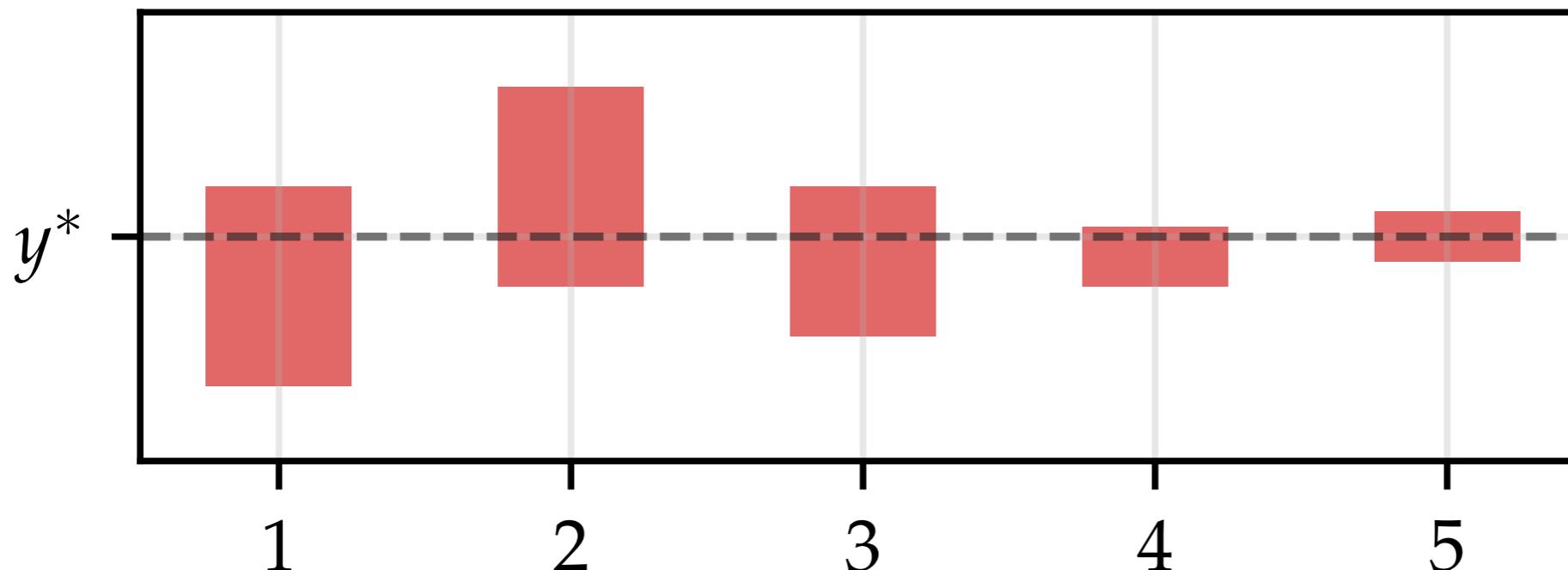
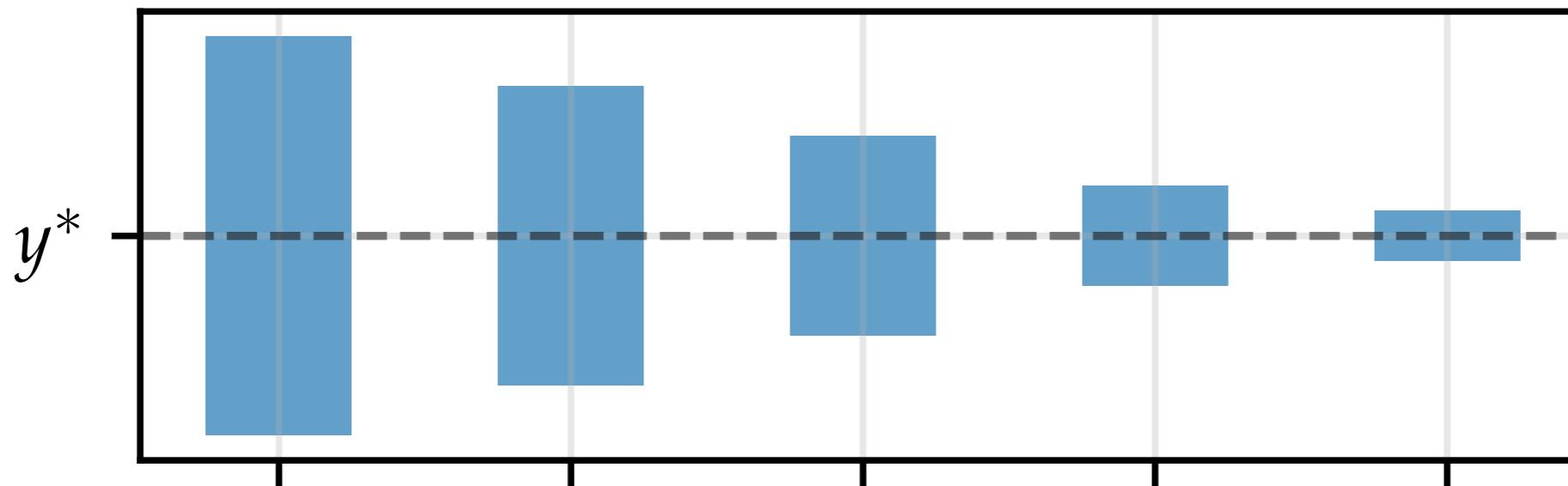
We want nested, non-increasing prediction intervals across exits.

consistency:

$$C_1(x) \subseteq C_2(x) \subseteq C_3(x)$$



$C_1(x) \ C_2(x) \ C_3(x) \ C_4(x) \ C_5(x)$



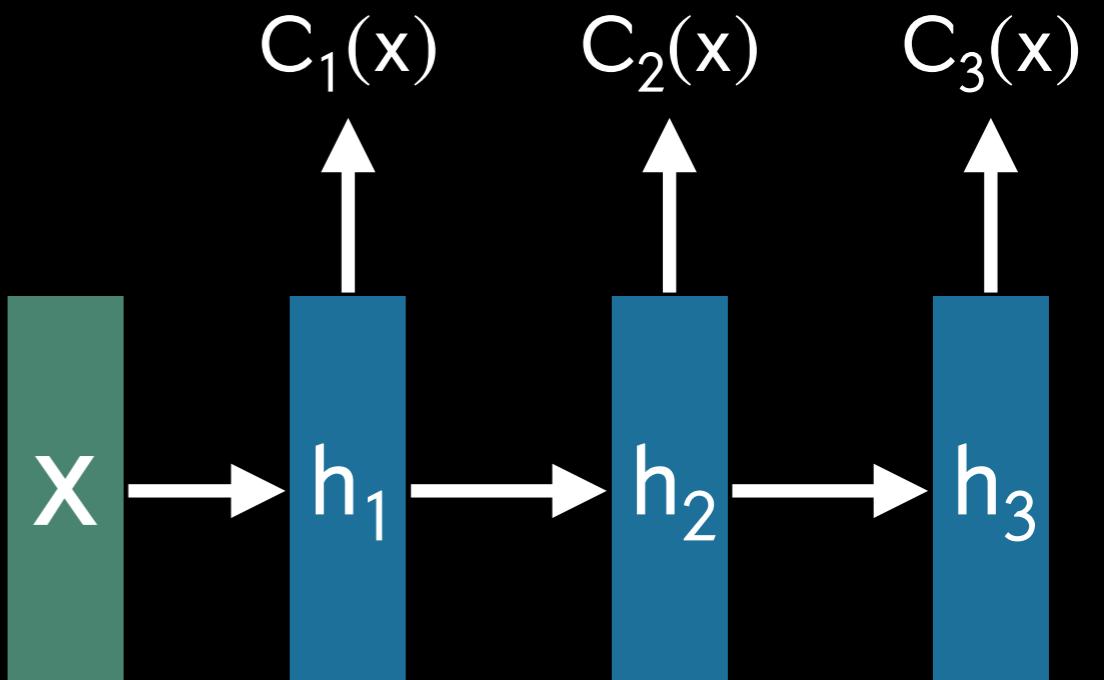
Early-Exit (t)

Anytime-Valid Confidence Sequences

We construct an *anytime-valid confidence sequence* across the exits.

$$P(\forall t, y^* \in C_t(x)) \geq 1 - \alpha$$

Due to approximations, we can only hope to achieve this for large datasets (and if y^ is from the training distribution).



Anytime-Valid Confidence Sequences

Derived from the following
predictive-likelihood martingale:

$$R_t(y) = \prod_{e=1}^t \frac{p_e(y | x, \mathcal{D})}{p_e(y | x, \hat{\theta}_e)} \quad \hat{\theta}_e \sim p(\theta_e | x, \mathcal{D})$$

Anytime-Valid Confidence Sequences

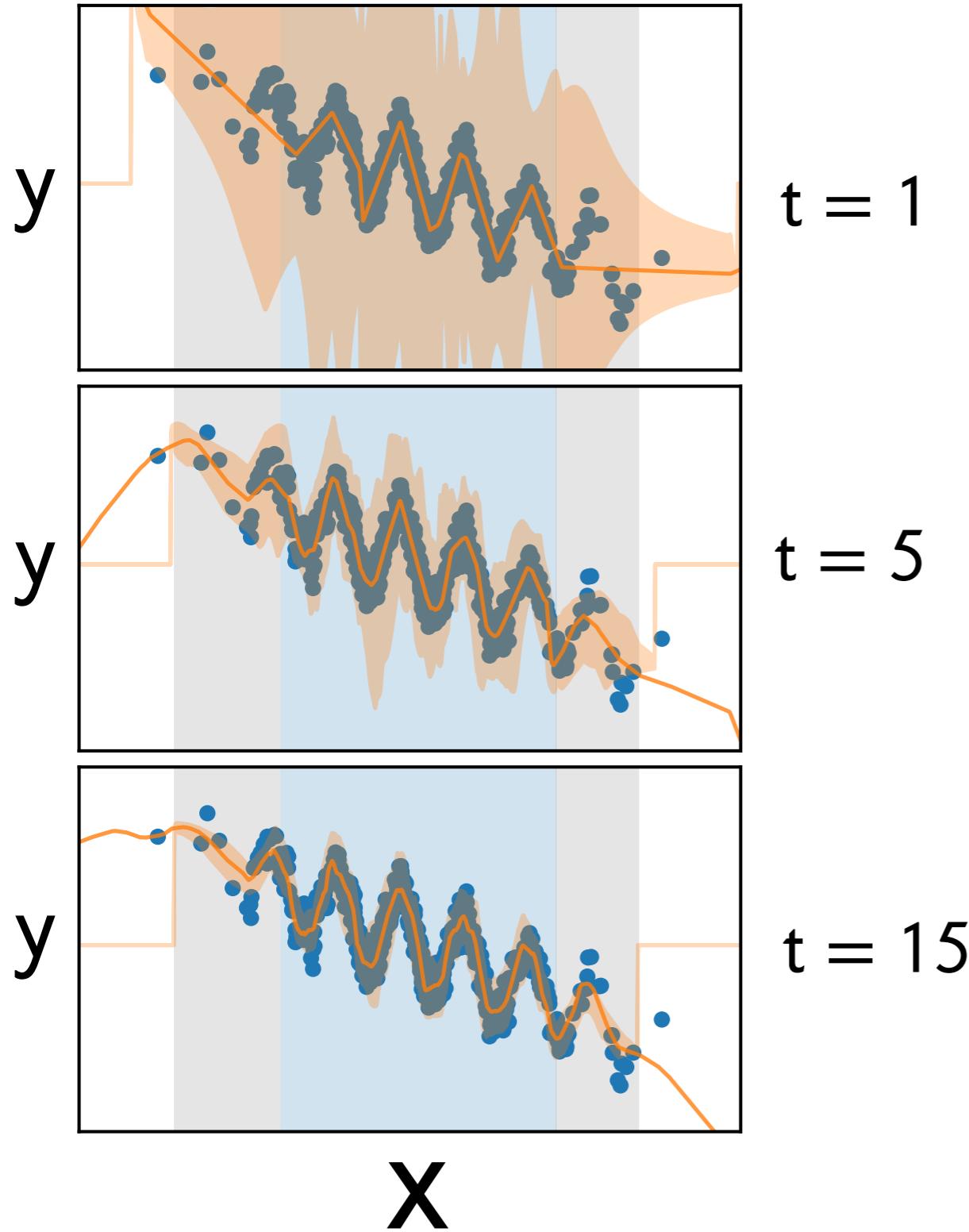
Derived from the following
predictive-likelihood martingale:

$$R_t(y) = \prod_{e=1}^t \frac{p_e(y | x, \mathcal{D})}{p_e(y | x, \hat{\theta}_e)} \quad \hat{\theta}_e \sim p(\theta_e | x, \mathcal{D})$$

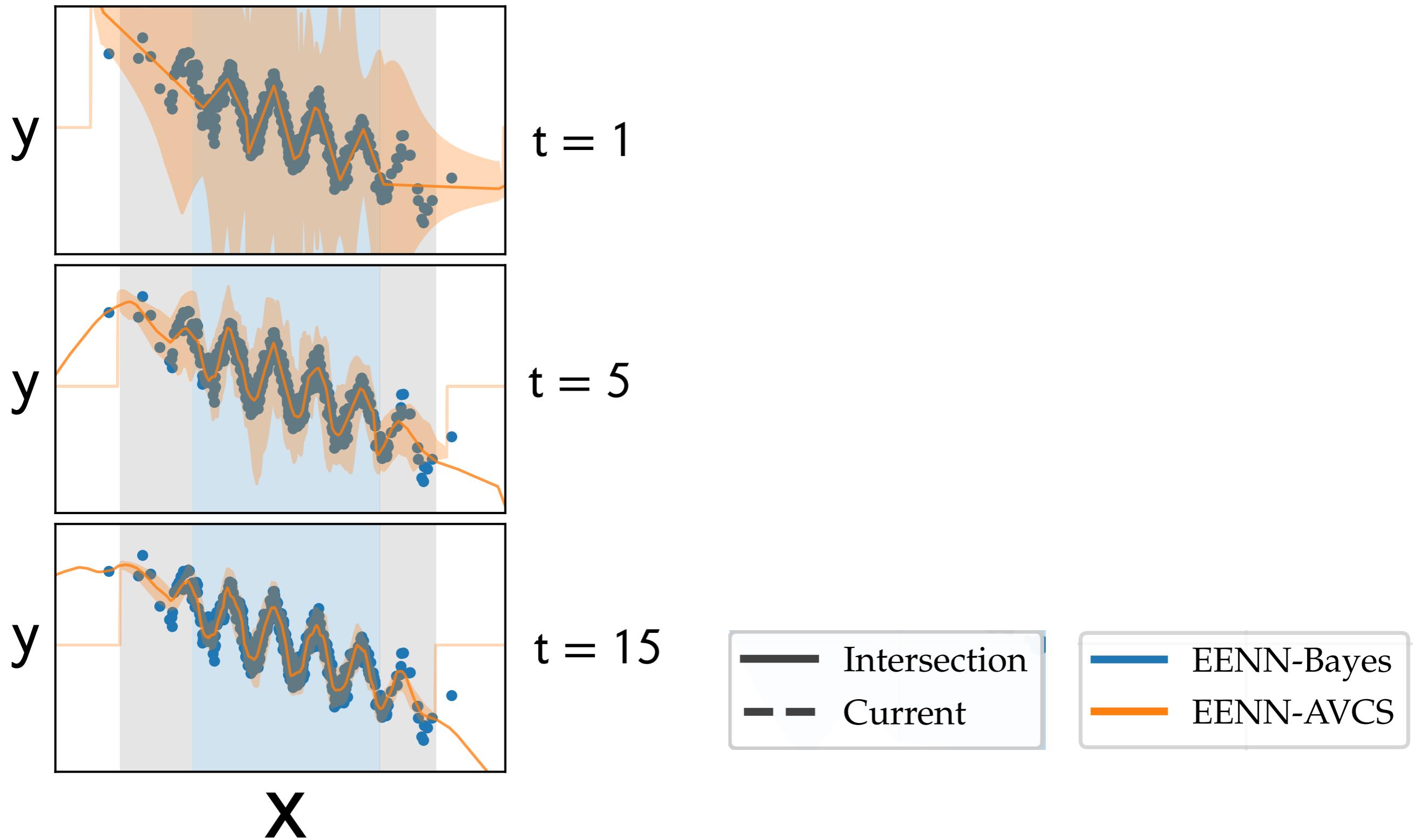
Construct set at time t as:

$$C_t(x) = \{y \in Y \mid R_t(y) \leq 1/\alpha\}$$

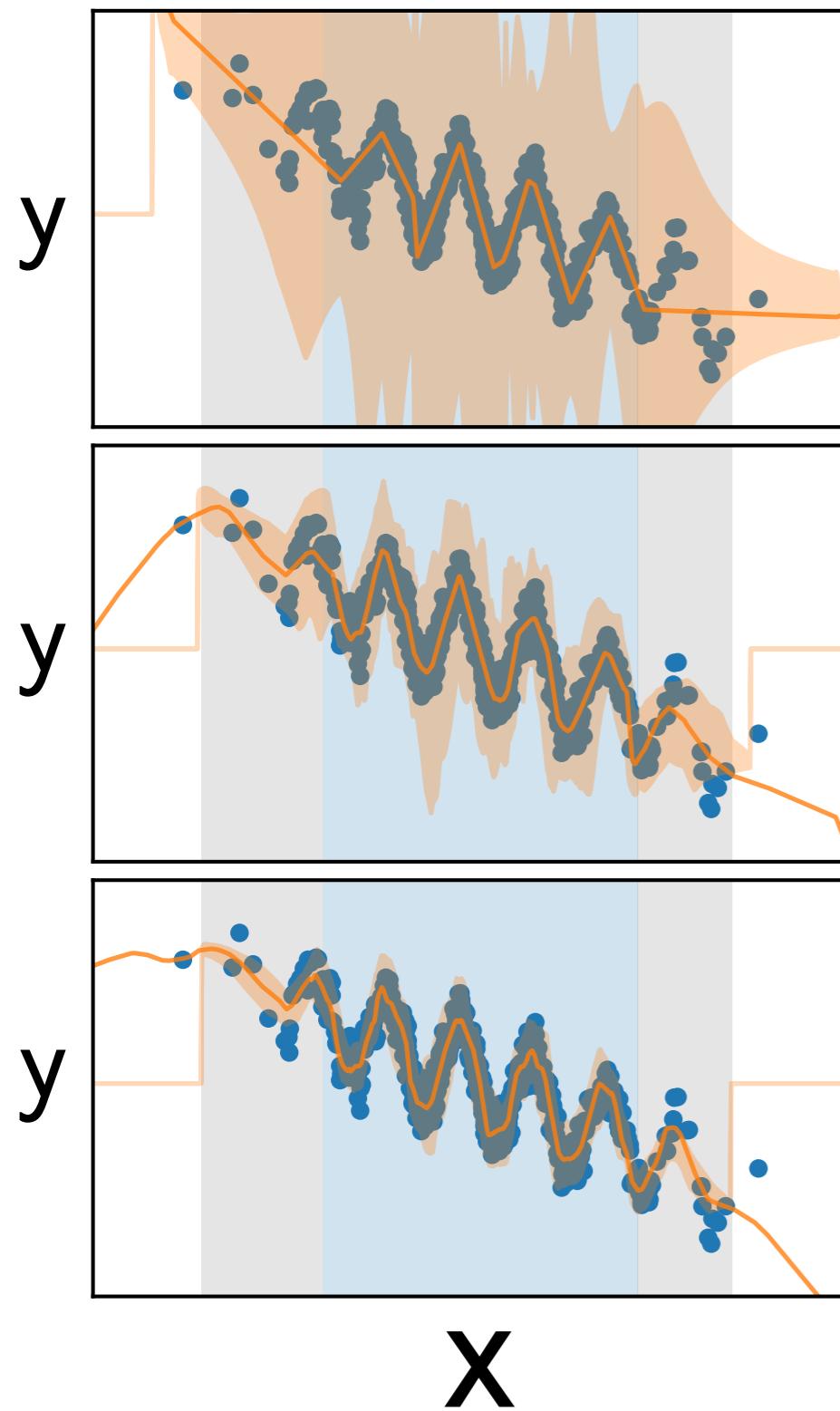
Regression Simulation



Regression Simulation



Regression Simulation

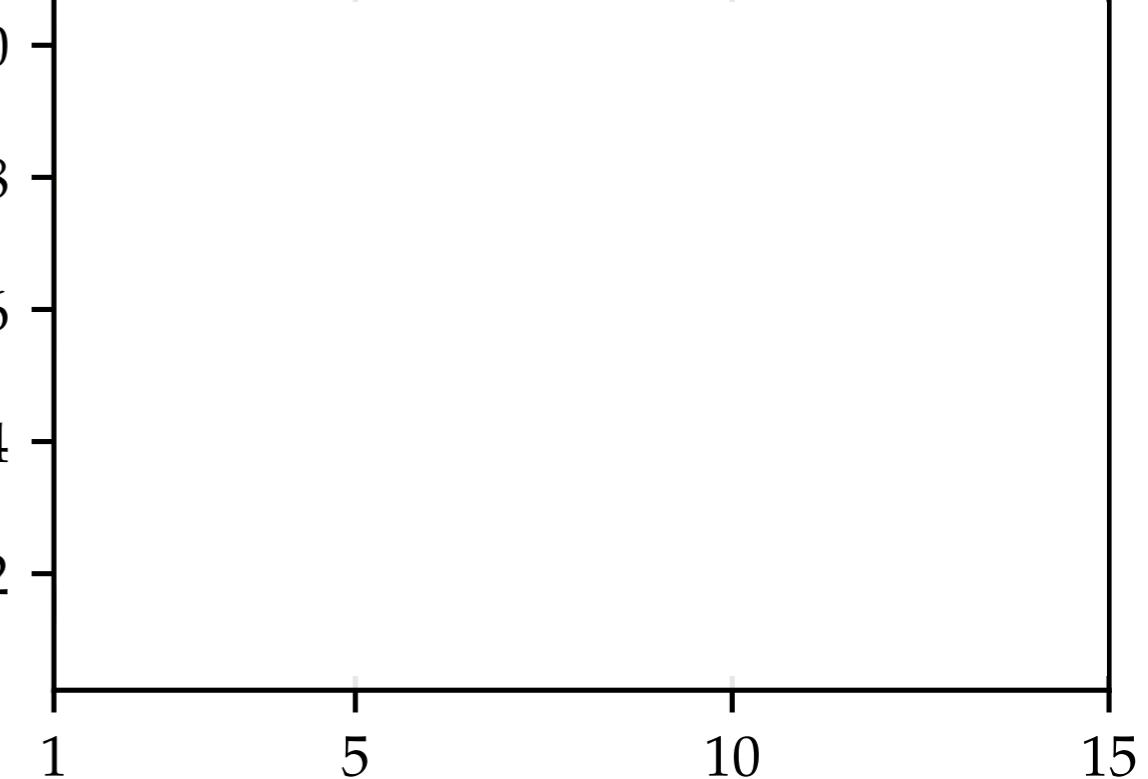


$t = 1$

$t = 5$

$t = 15$

Average Interval Size

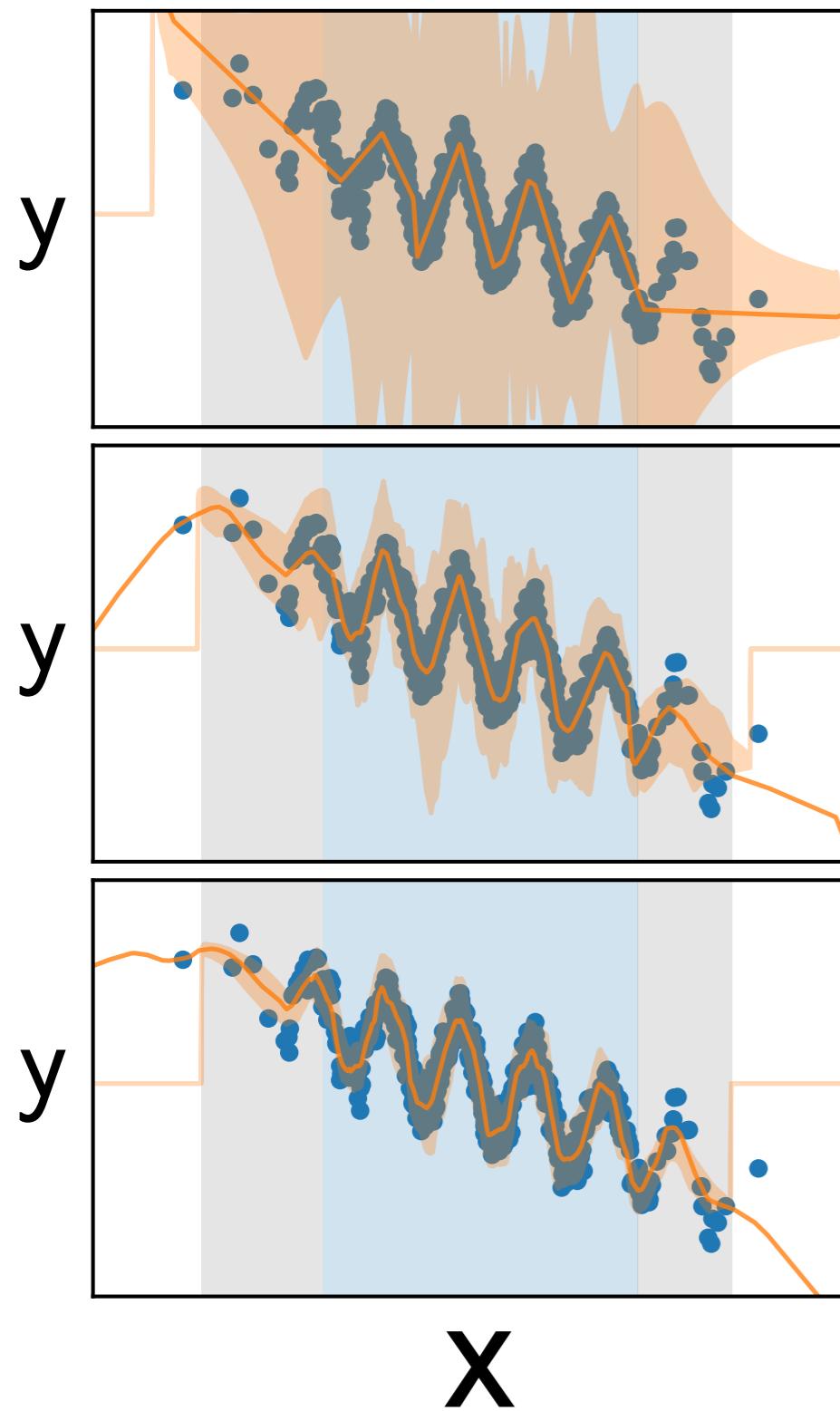


Time / Early-Exit

Intersection
Current

EENN-Bayes
EENN-AVCS

Regression Simulation

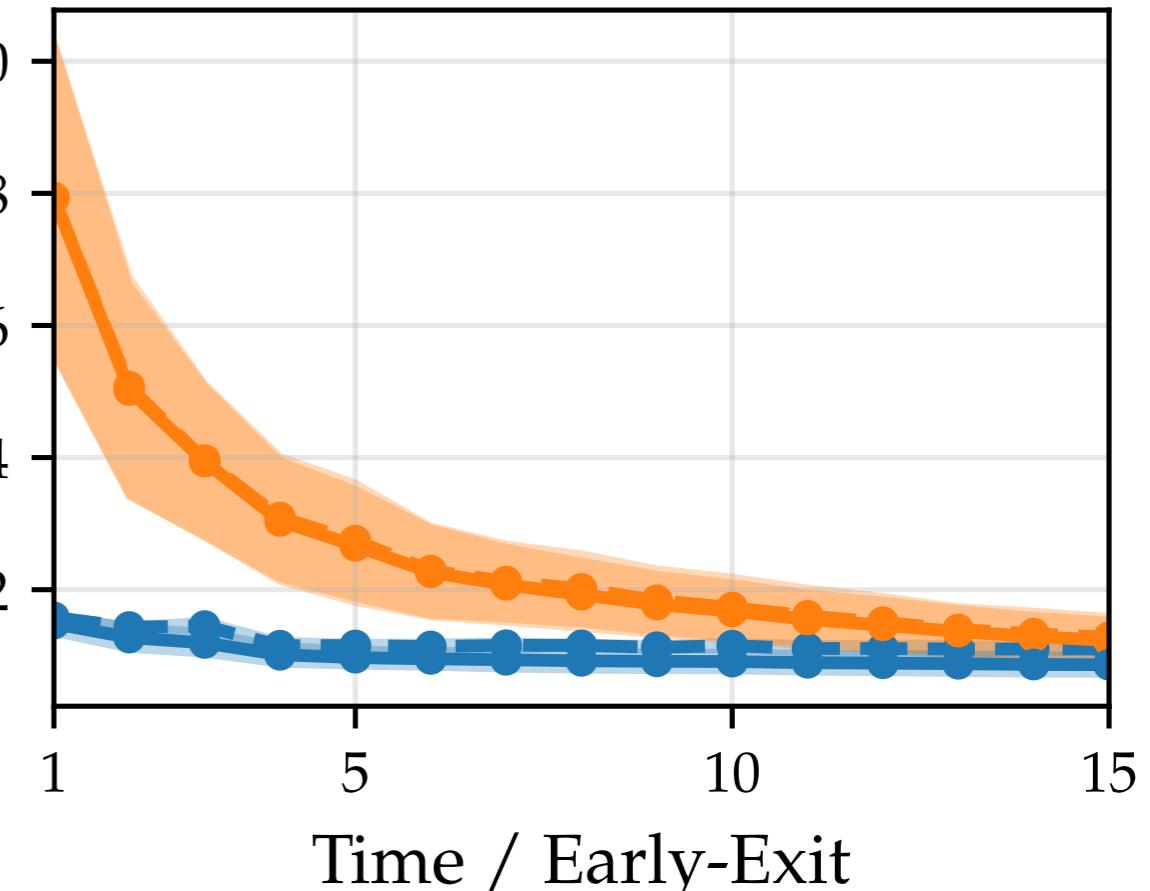


$t = 1$

$t = 5$

$t = 15$

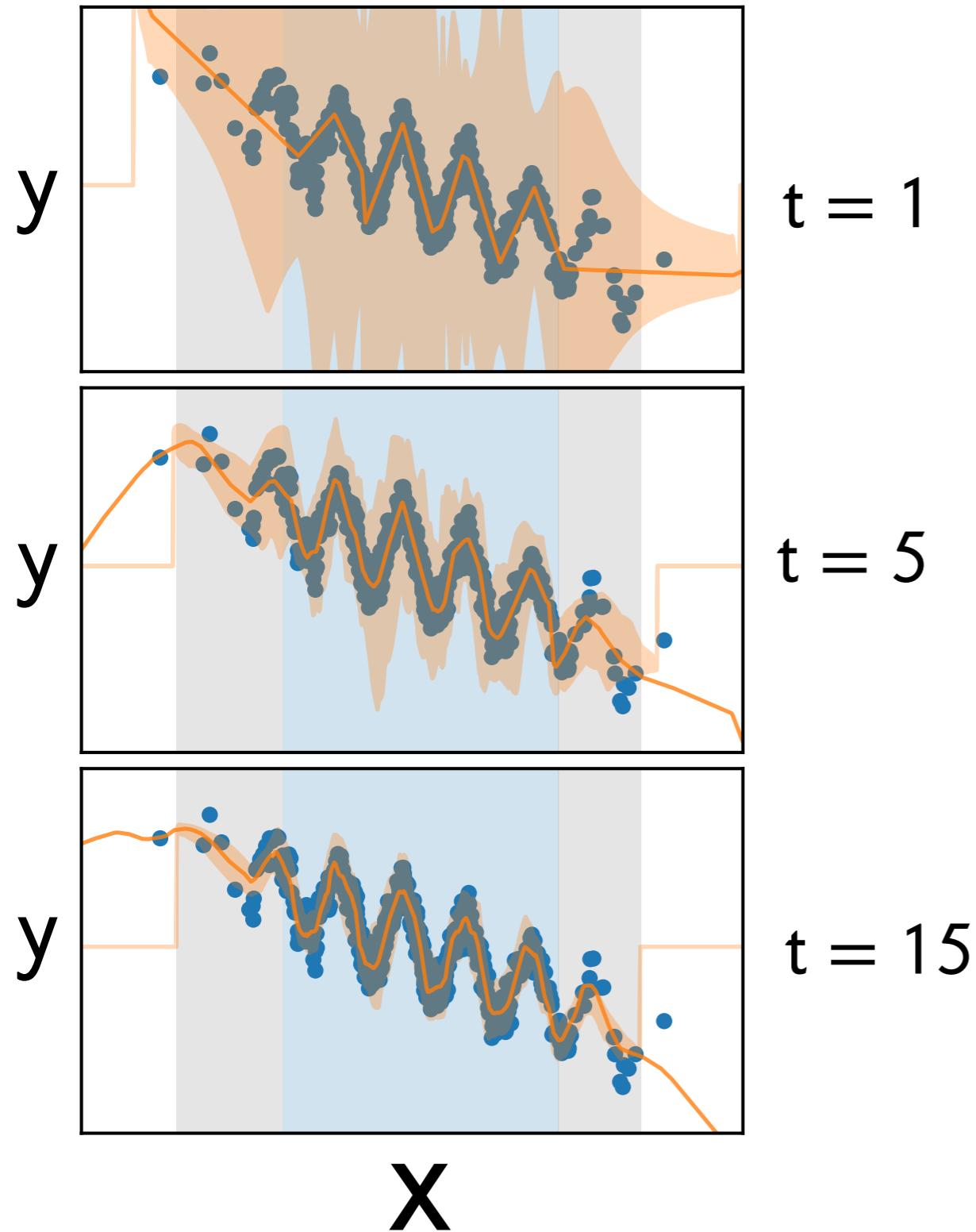
Average Interval Size



— Intersection
- - Current

— EENN-Bayes
— EENN-AVCS

Regression Simulation



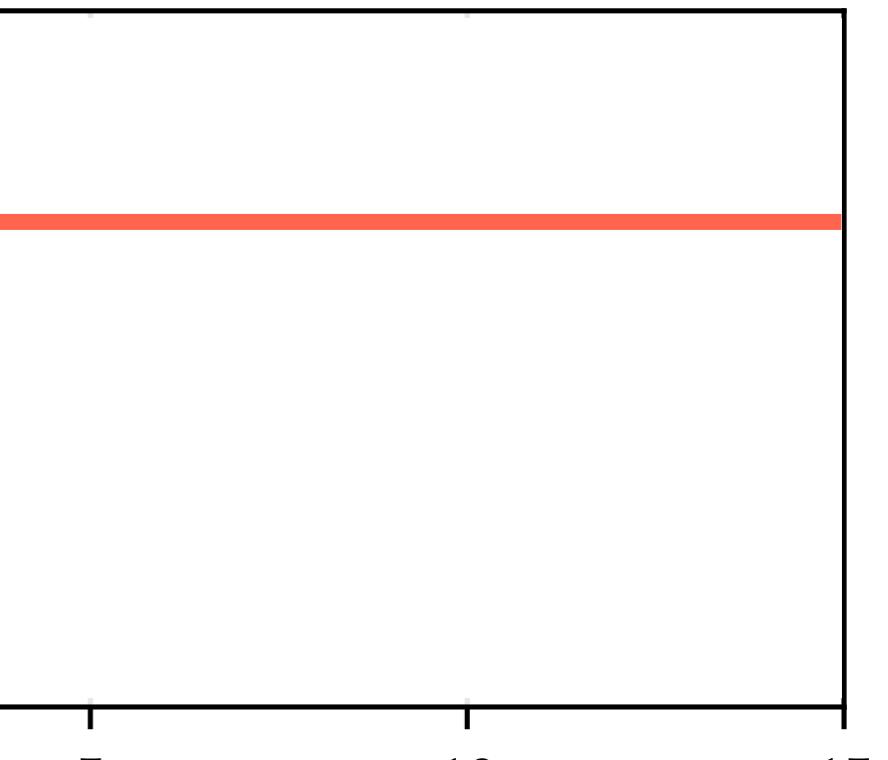
$t = 1$

$t = 5$

$t = 15$

Marginal Coverage

105

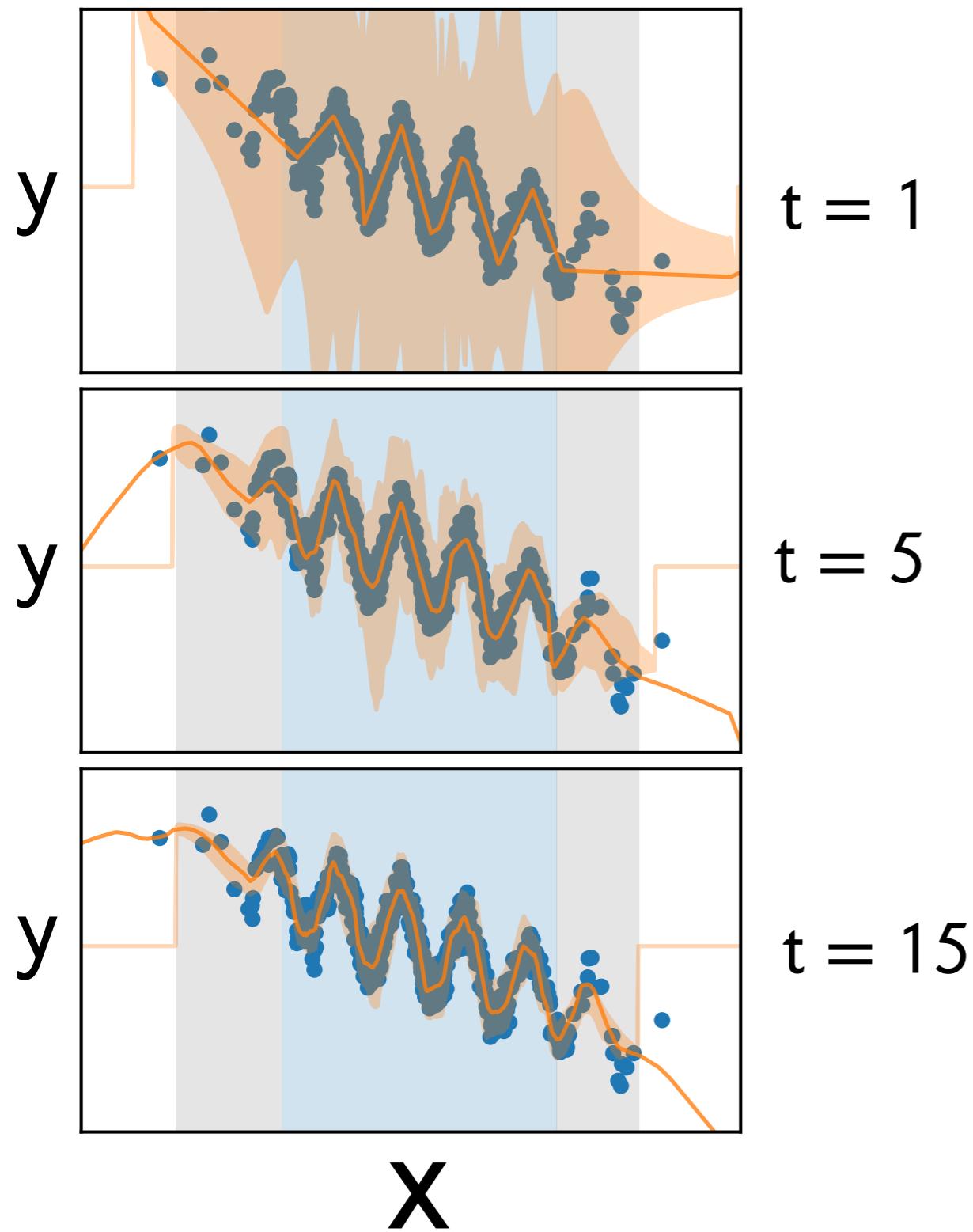


Time / Early-Exit

Intersection
Current

EENN-Bayes
EENN-AVCS

Regression Simulation

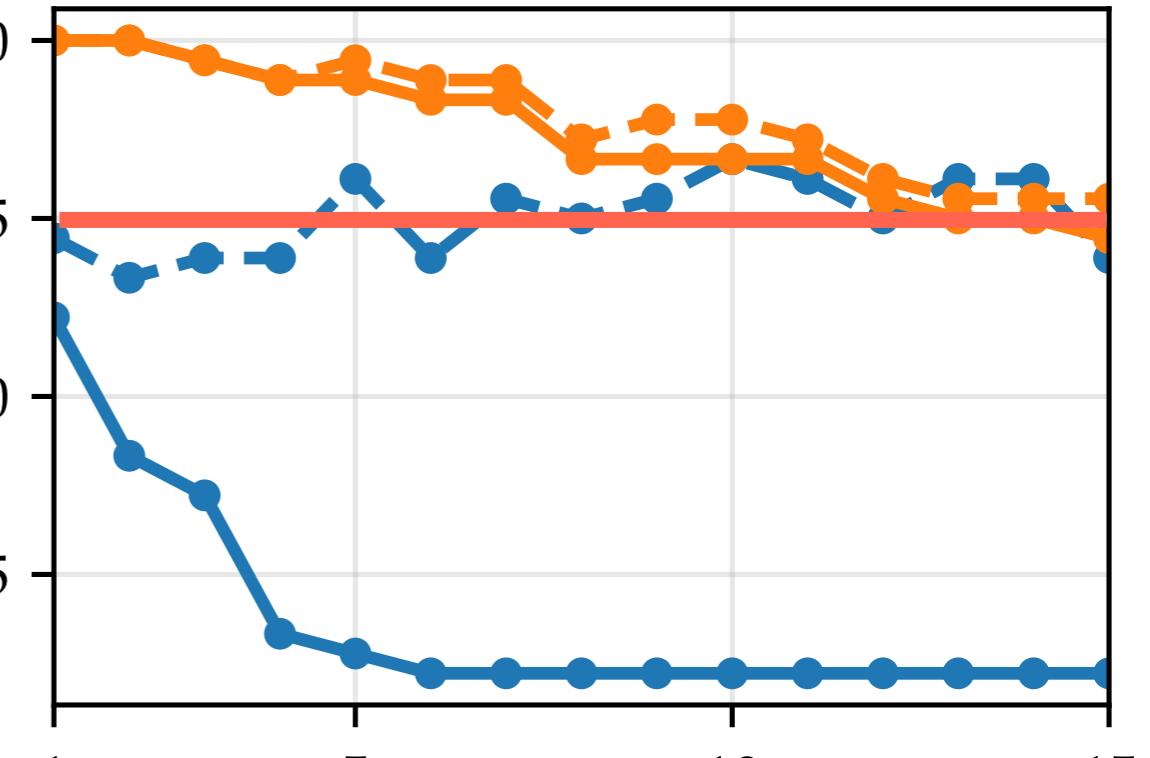


$t = 1$

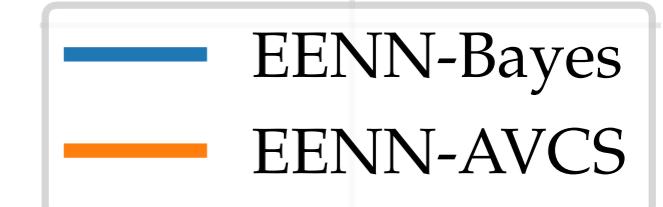
$t = 5$

$t = 15$

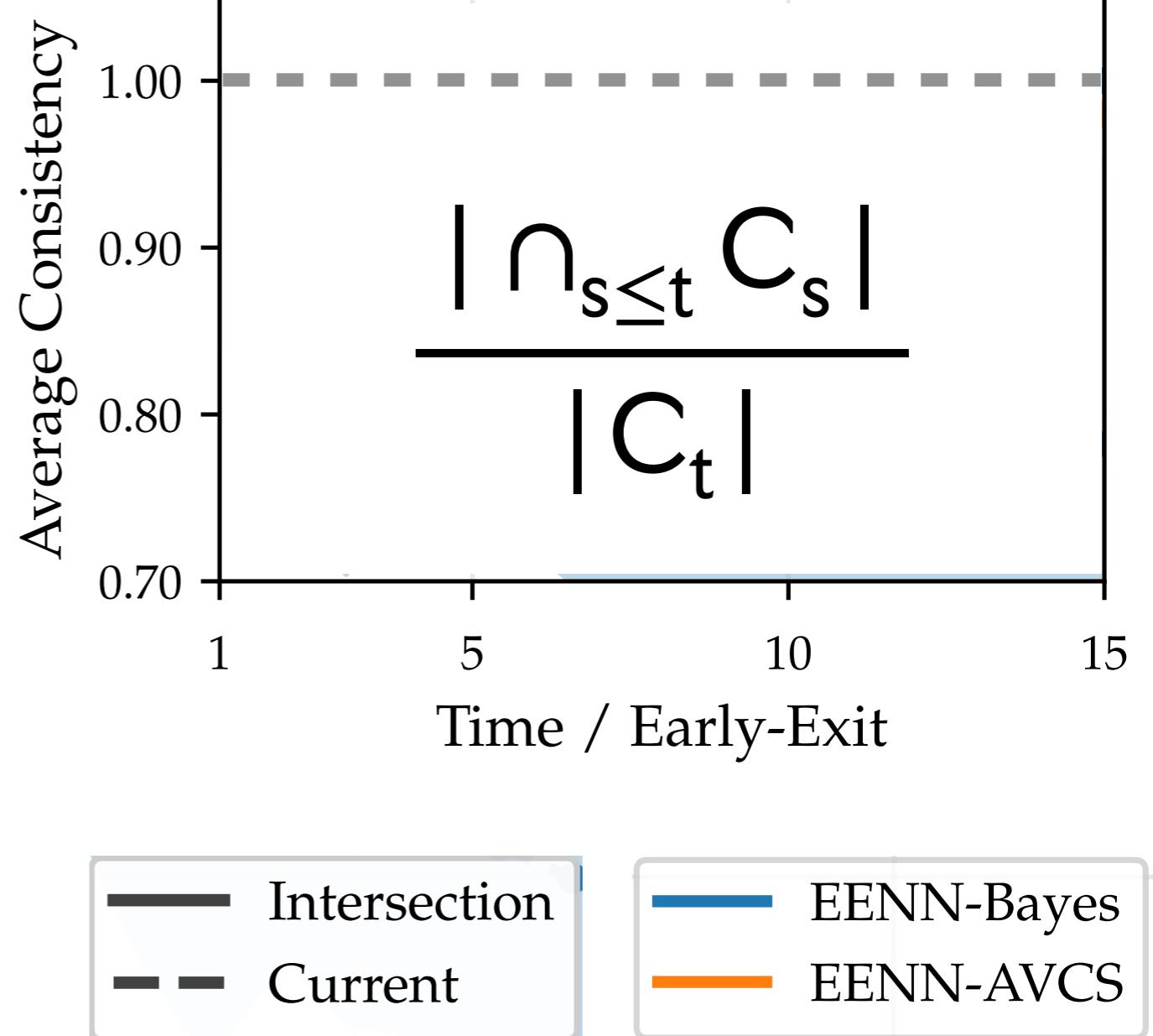
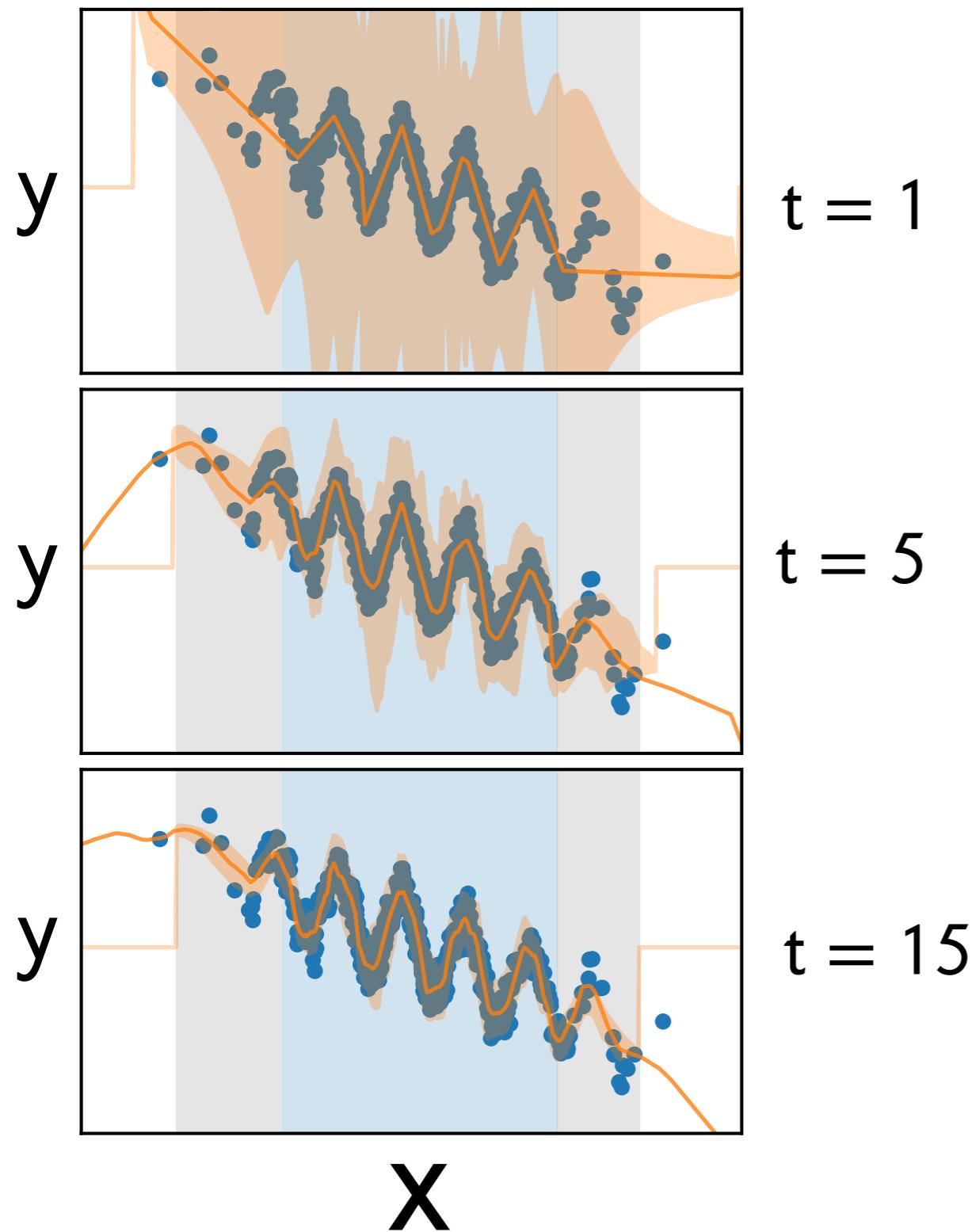
Marginal Coverage



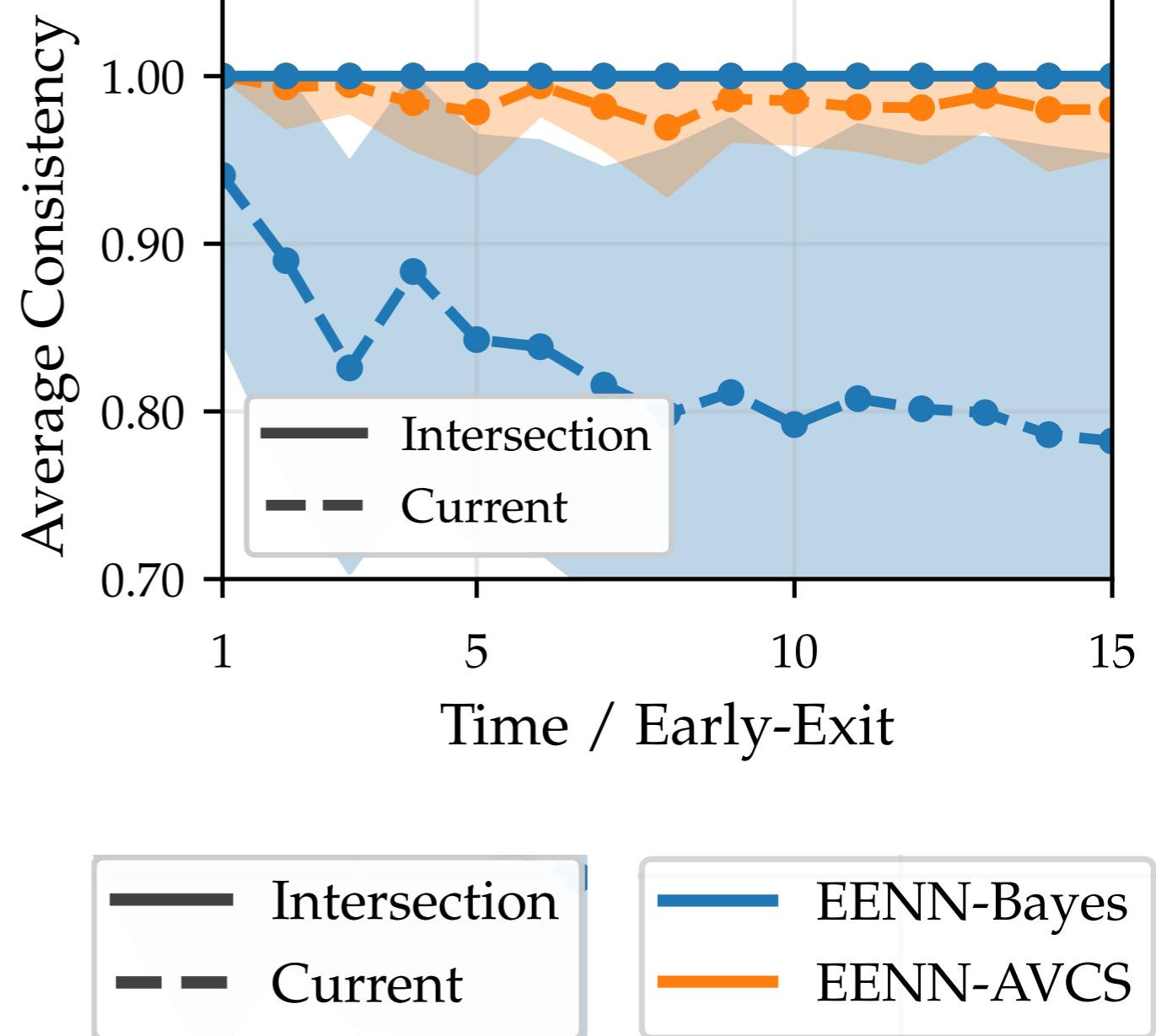
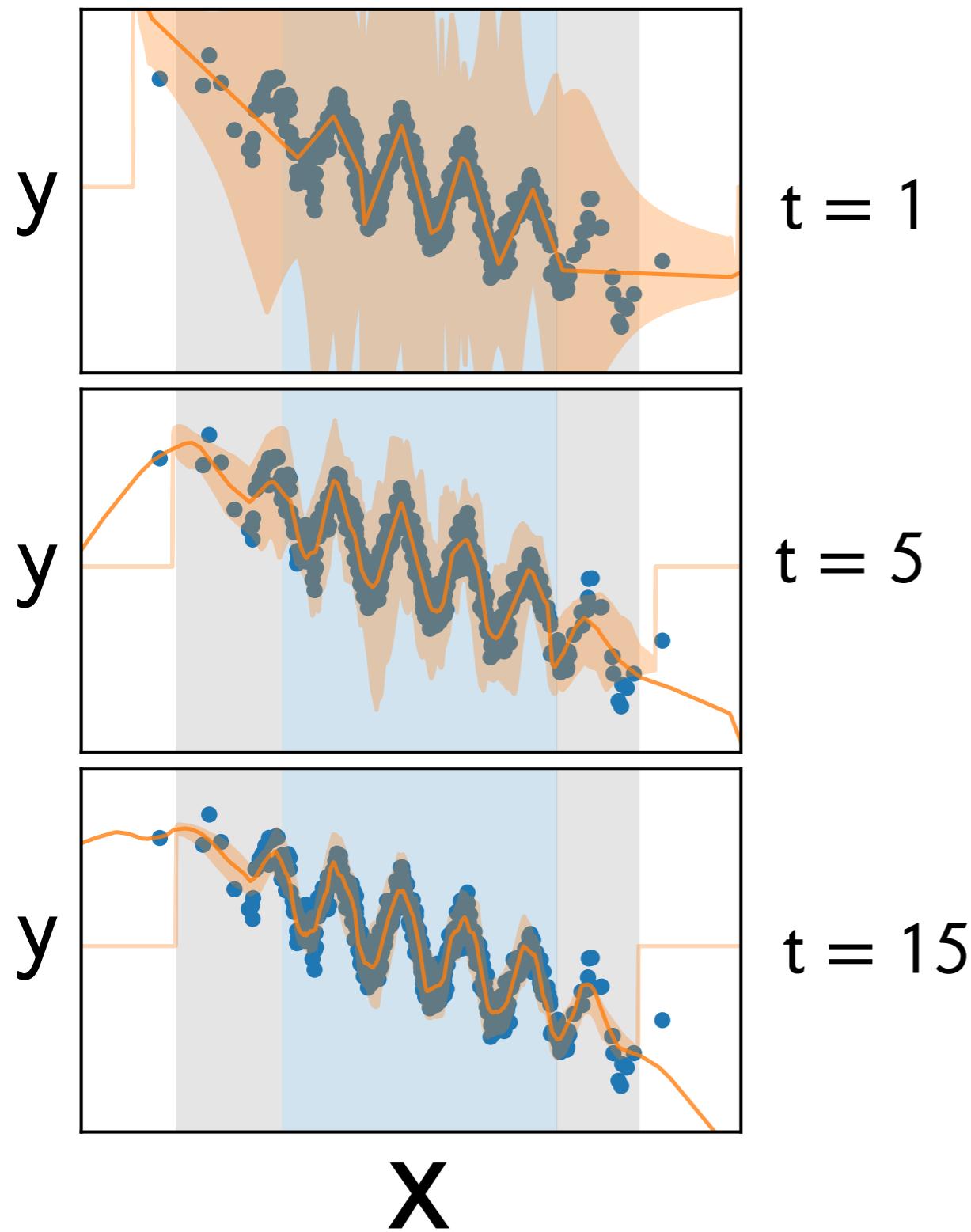
Time / Early-Exit



Regression Simulation



Regression Simulation

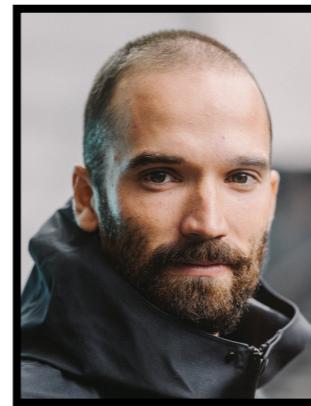


Summary

- ⊗ Early-exit neural networks have mostly marginal anytime properties (and overthink)
 - ⊗ We give them better conditional monotonicity via a product ensemble.
-
- ⊗ Also want consistency in predictive uncertainty across exits.
 - ⊗ We enforce this with anytime-valid confidence sequences.

Thank you! Questions?

paper



Metod
Jazbec



James U.
Allingham



UvA - BOSCH

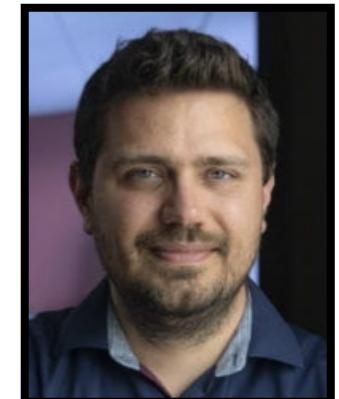
DELTA LAB



Dan
Zhang



Patrick
Forré



Stephan
Mandt