

New and Future Extensions to Cobweb

Christopher J. MacLellan
Georgia Institute of Technology



Exciting and Disruptive Times!

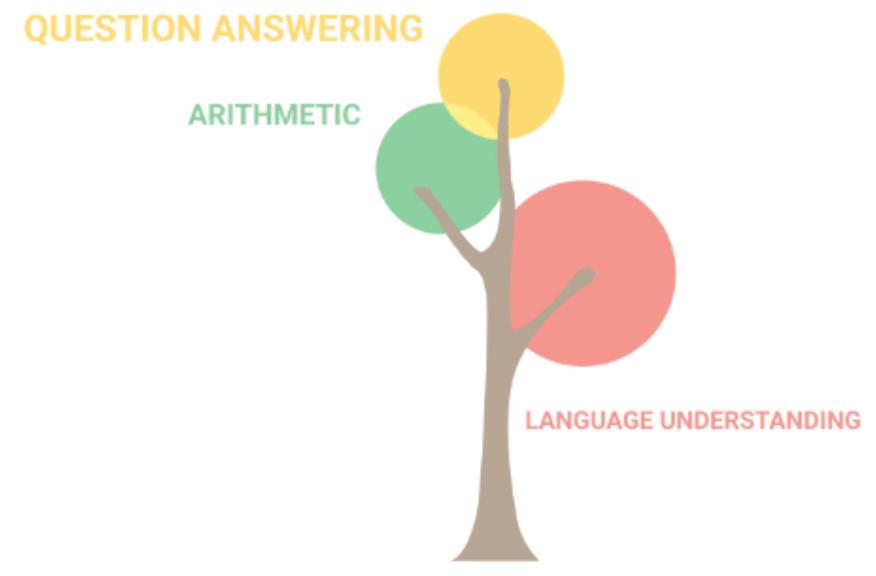
Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck Varun Chandrasekaran Ronen Eldan Johannes Gehrke
Eric Horvitz Ece Kamar Peter Lee Yin Tat Lee Yuanzhi Li Scott Lundberg
Harsha Nori Hamid Palangi Marco Tulio Ribeiro Yi Zhang

Microsoft Research

Abstract

Artificial intelligence (AI) researchers have been developing and refining large language models (LLMs) that exhibit remarkable capabilities across a variety of domains and tasks, challenging our understanding of learning and cognition. The latest model developed by OpenAI, GPT-4 [Ope23], was trained using an unprecedented scale of compute and data. In this paper, we report on our investigation of an early version of GPT-4, when it was still in active development by OpenAI. We contend that (this early version of) GPT-4 is part of a new cohort of LLMs (along with ChatGPT and Google's PaLM for example) that exhibit more general intelligence than previous AI models. We discuss the rising capabilities and implications of these models. We demonstrate that, beyond its mastery of language, GPT-4 can solve novel and difficult tasks that span mathematics, coding, vision, medicine, law, psychology and more, without needing any special prompting. Moreover, in all of these tasks, GPT-4's performance is strikingly close to human-level performance, and often vastly surpasses prior models such as ChatGPT. Given the breadth and depth of GPT-4's capabilities, we believe that it could reasonably be viewed as an early (yet still incomplete) version of an artificial general intelligence (AGI) system. In our exploration of GPT-4, we put special emphasis on discovering its limitations, and we discuss the challenges ahead for advancing towards deeper and more comprehensive versions of AGI, including the possible need for pursuing a new paradigm that moves beyond next-word prediction. We conclude with reflections on societal influences of the recent technological leap and future research directions.



8 billion parameters

Source: <https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html>

Some Human-like Performance

...but not Human-Like Learning

Substantial Data and Compute Requirements

- Require internet scale data
- Cost millions of dollars to train (>\$10M for large models)
- Large power / carbon footprints for both training and inference (training > hundreds of households annual power)

Metrics from Stanford AI Index Report 2023

Difficult to Adapt and Update with New Data

- Primarily support offline batch training (rather than incremental, continual learning)
- Exhibit catastrophic forgetting of old knowledge when trained on new data

If you are interested in Human-Like Learning, then consider checking out the upcoming AAAI Spring Symposium on the topic at <https://humanlikelearning.com>

Examples of Opportunities for Human-Like Learning

Missing Knowledge

Model: GPT-4

CM Can you give me advice about how to play the game Space Transit, created by a research group at UCSD?

 I'm sorry, but as my knowledge is limited to updates until September 2021, I am not familiar with the game "Space Transit" created by a research group at UCSD. If this game was developed after September 2021, I won't be able to provide specific advice about it. However, if you have any questions about general gaming strategies, game design, or any other related topics, I'd be more than happy to help.

Incorrect Knowledge

Model: GPT-4

CM Please give me a bio about Christopher J. MacLellan, who is an assistant professor at Georgia Tech's School of Interactive Computing.

 Christopher J. MacLellan is an accomplished Assistant Professor in the School of Interactive Computing at the Georgia Institute of Technology (Georgia Tech). With a strong background in artificial intelligence, machine learning, and human-computer interaction, his research focuses on developing intelligent systems that can augment human learning and creativity by leveraging collaboration between humans and AI.

Dr. MacLellan earned his Bachelor's degree in Computer Science from the University of California, Santa Cruz, before pursuing his Master's and Ph.D. in Human-Computer Interaction from Carnegie Mellon University. During his time at Carnegie Mellon, Christopher contributed significantly to the field of AI and learning sciences, receiving numerous accolades for his innovative research.

As an assistant professor at Georgia Tech, Dr. MacLellan is dedicated to advancing the field of interactive computing and exploring novel ways to empower individuals through the use of AI. His interdisciplinary approach has led to notable collaborations, including the development of intelligent tutoring systems, tools for enhancing the creative process, and techniques for improving the usability of AI systems.

Future Directions for Cobweb Research

Performance

- Explore new learning and performance mechanisms
- Translate ideas about input processing from deep learning to Cobweb (convolutions, attention, etc.)
- Increase scale of training data

Efficiency / Adaptability

- Improve asymptotic runtime of core approach
- Investigate ability to support continual, incremental learning without forgetting
- Explore data and power efficiency relative to deep learning techniques

Changes to Core Cobweb Mechanisms

Information Theory Variant of Category Utility

Expected information gained, rather than expected correct guesses

- Probability Theoretic Category Utility

$$CU(C_k) = P(C_k) \sum_i \sum_j \left[P(A_i = v_{ij} | C_k)^2 - P(A_i = v_{ij})^2 \right]$$

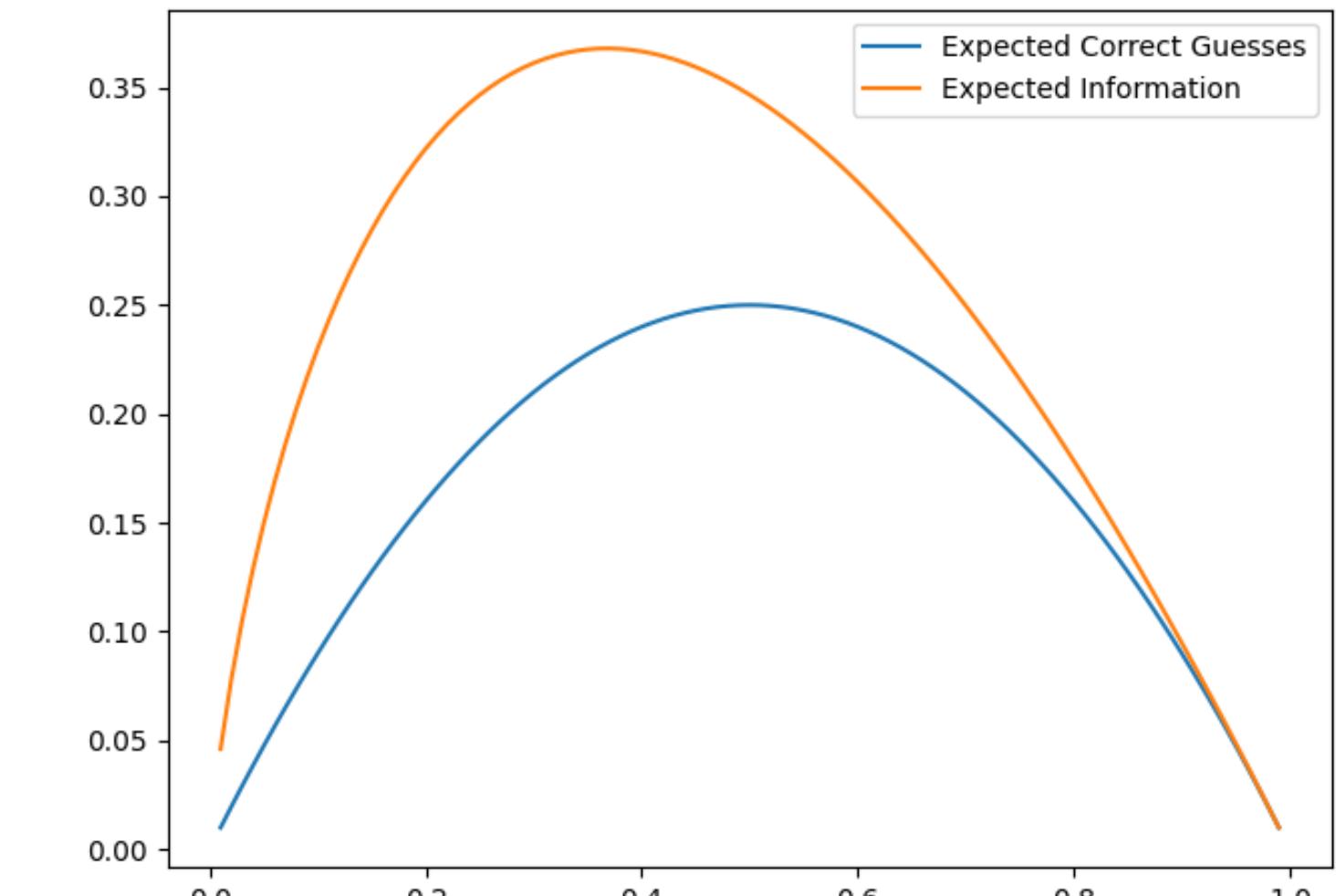
- Information Theoretic Category Utility – i.e., Mutual Information (MI)

$$MI(C_k) = P(C_k) \sum_i [H(A_i = V) - H(A_i = V | C_k)],$$

where $H(A_i = V) = - \sum_j P(A_i = v_{ij}) \times \ln(P(A_i = v_{ij}))$
 (i.e., Entropy)

- Information Theoretic Partition Utility:

$$PU(C_1, \dots, C_m) = \sum_{k=1}^m \frac{MI(C_k)}{m}$$



$$\begin{aligned}
 H(A_i = V) &= - \sum_j (p_{ij} \times \ln(p_{ij})) = - \sum_j p_{ij} \times \ln(1 - (1 - p_{ij})) \\
 &= - \sum_j p_{ij} \left[-(1 - p_{ij}) - \frac{1}{2}(1 - p_{ij})^2 - \frac{1}{3}(1 - p_{ij})^3 + \dots \right] \quad (\text{by series expansion}) \\
 &= \boxed{\sum_j p_{ij}} \left[(1 - p_{ij}) + \frac{1}{2}(1 - p_{ij})^2 + \frac{1}{3}(1 - p_{ij})^3 + \dots \right] \\
 &\rightarrow \sum_j p_{ij} \times (1 - p_{ij}) = \sum_j p_{ij} - \sum_j p_{ij}^2 = 1 - \boxed{\sum_j p_{ij}^2}
 \end{aligned}$$

The correct guesses of an attribute-value probability is the first order approximation of the entropy, so CU is an approximation of MI.

Information Theory Variant of Category Utility

Expected information gained, rather than expected correct guesses

- Probability Theoretic Category Utility

$$CU(C_k) = P(C_k) \sum_i \sum_j \left[P(A_i = v_{ij} | C_k)^2 - P(A_i = v_{ij})^2 \right]$$

- Information Theoretic Category Utility: Mutual Information (MI)
- Another advantage of using information theory is that most distributions (e.g., categorical and normal) have closed-form definitions for Entropy.**

$$\text{where } H(A_i = V) = - \sum_j P(A_i = v_{ij}) \times \ln(P(A_i = v_{ij}))$$

- Information Theoretic Partition Utility:

$$PU(C_1, \dots, C_m) = \sum_{k=1}^m \frac{MI(C_k)}{m}$$



$$\begin{aligned}
 H(A_i = V) &= - \sum_j p_{ij} \times \ln(1 - (1 - p_{ij})) \\
 &= - \sum_j p_{ij} \left[-(1 - p_{ij}) - \frac{1}{2}(1 - p_{ij})^2 - \frac{1}{3}(1 - p_{ij})^3 + \dots \right] \quad (\text{by series expansion}) \\
 &= \sum_j p_{ij} \left[(1 - p_{ij}) + \frac{1}{2}(1 - p_{ij})^2 + \frac{1}{3}(1 - p_{ij})^3 + \dots \right]
 \end{aligned}$$

$\sum_j p_{ij} \times (1 - p_{ij}) = \sum_j p_{ij} - \sum_j p_{ij}^2 = 1 - \boxed{\sum_j p_{ij}^2}$

The correct guesses of an attribute-value probability is the first order approximation of the entropy, so CU is an approximation of MI.

Attribute Smoothing

Ensures well defined behavior when estimating from a single example

Nominal Attribute Smoothing

$$H(A_i = V_i | j) = - \sum_j p_{ij} \times \ln(p_{ij}),$$

where $p_{ij} = \frac{n_{ij} + \alpha}{n_i + \alpha \times d}$, α is a smoothing parameter (a small positive value), and d is the number of possible values of attribute A_i .

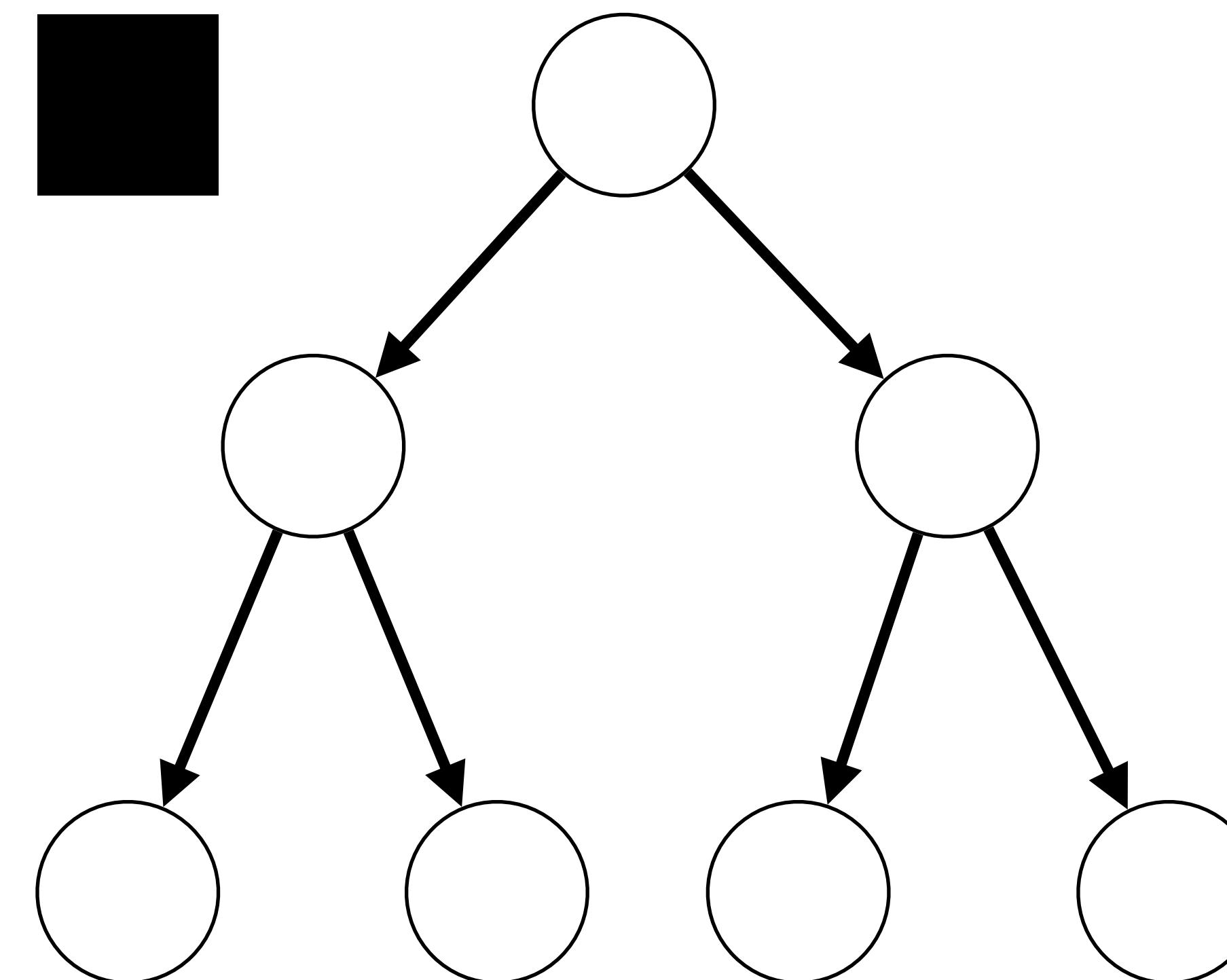
Continuous Attribute Smoothing

$$H(A_i = V) = \frac{1}{2} \ln \left[2\pi \left(\sigma_i^2 + \sigma_{acuity}^2 \right) \right] + \frac{1}{2},$$

where σ_i is the sample standard deviation and σ_{acuity} is a smoothing parameter (a small positive value).

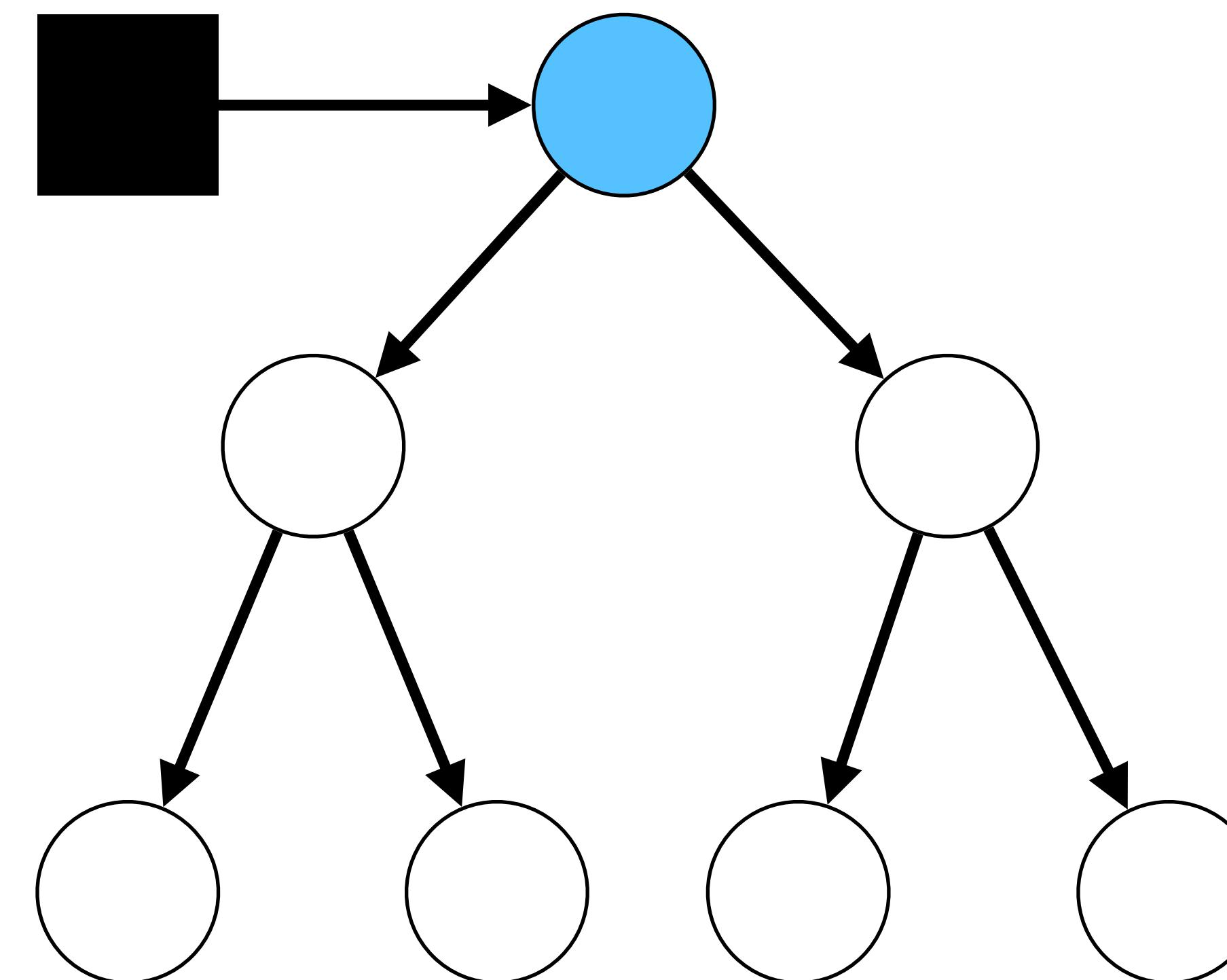
Original Categorization Approach

Greedy search and prediction from a single node



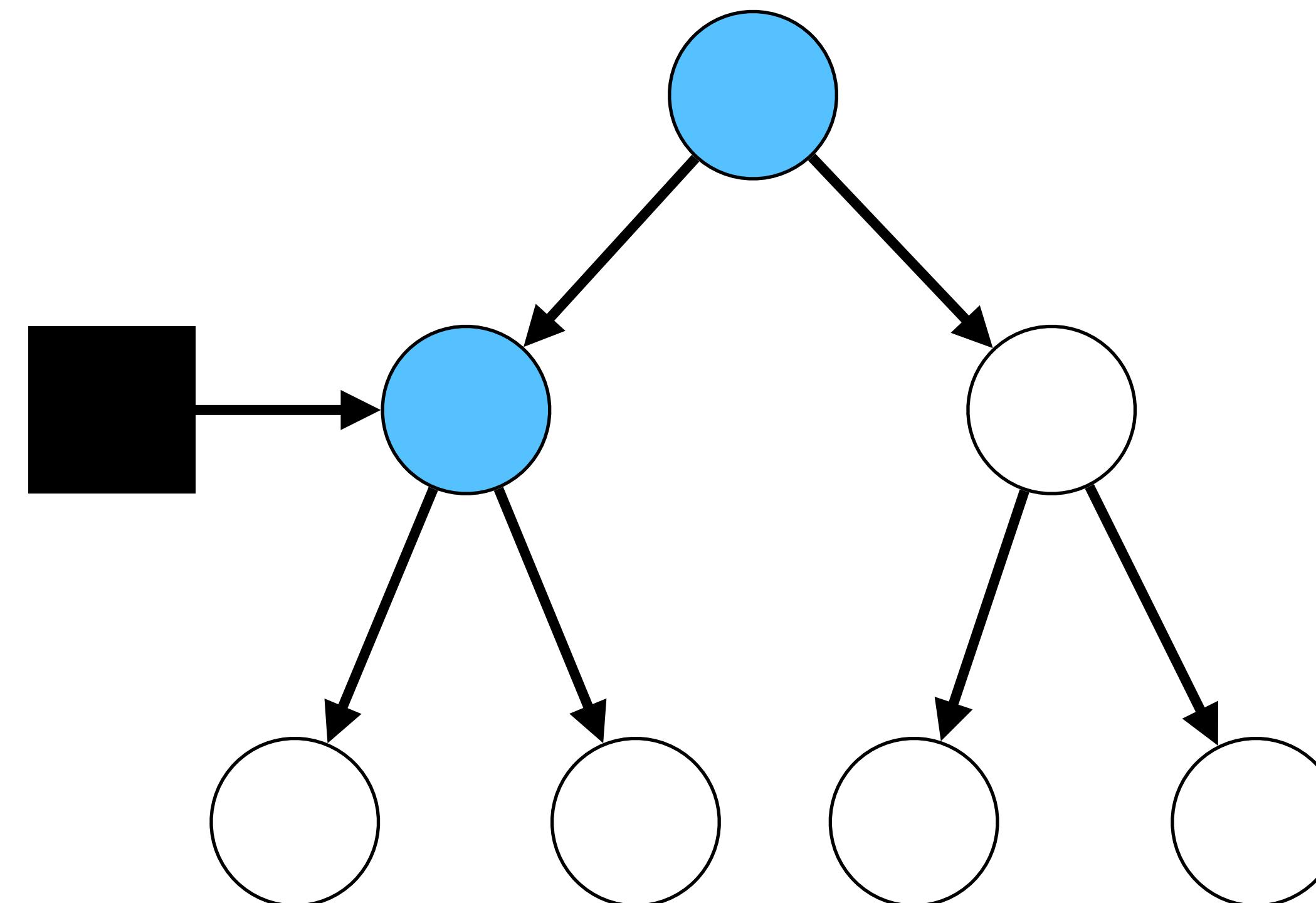
Original Categorization Approach

Greedy search and prediction from a single node



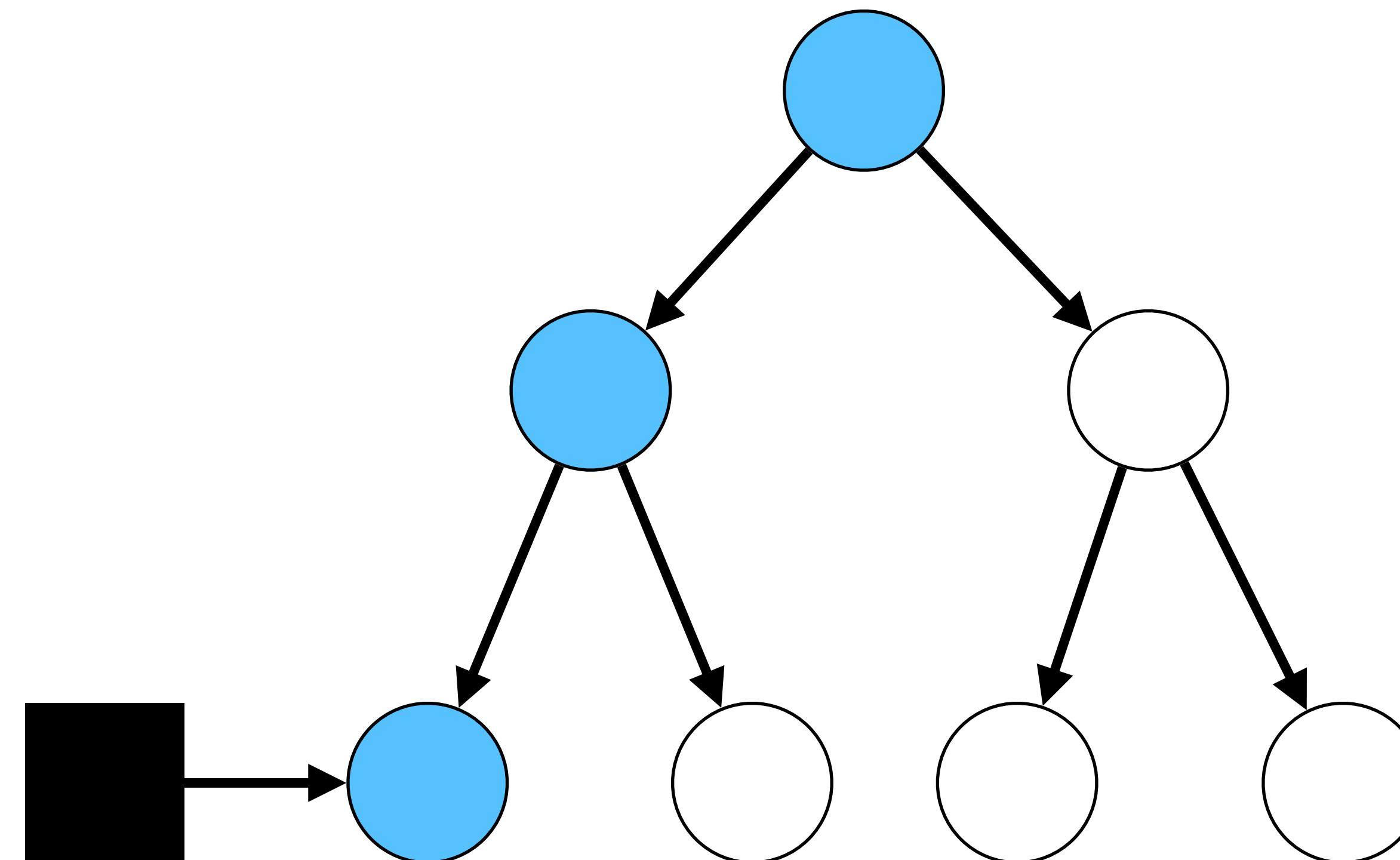
Original Categorization Approach

Greedy search and prediction from a single node



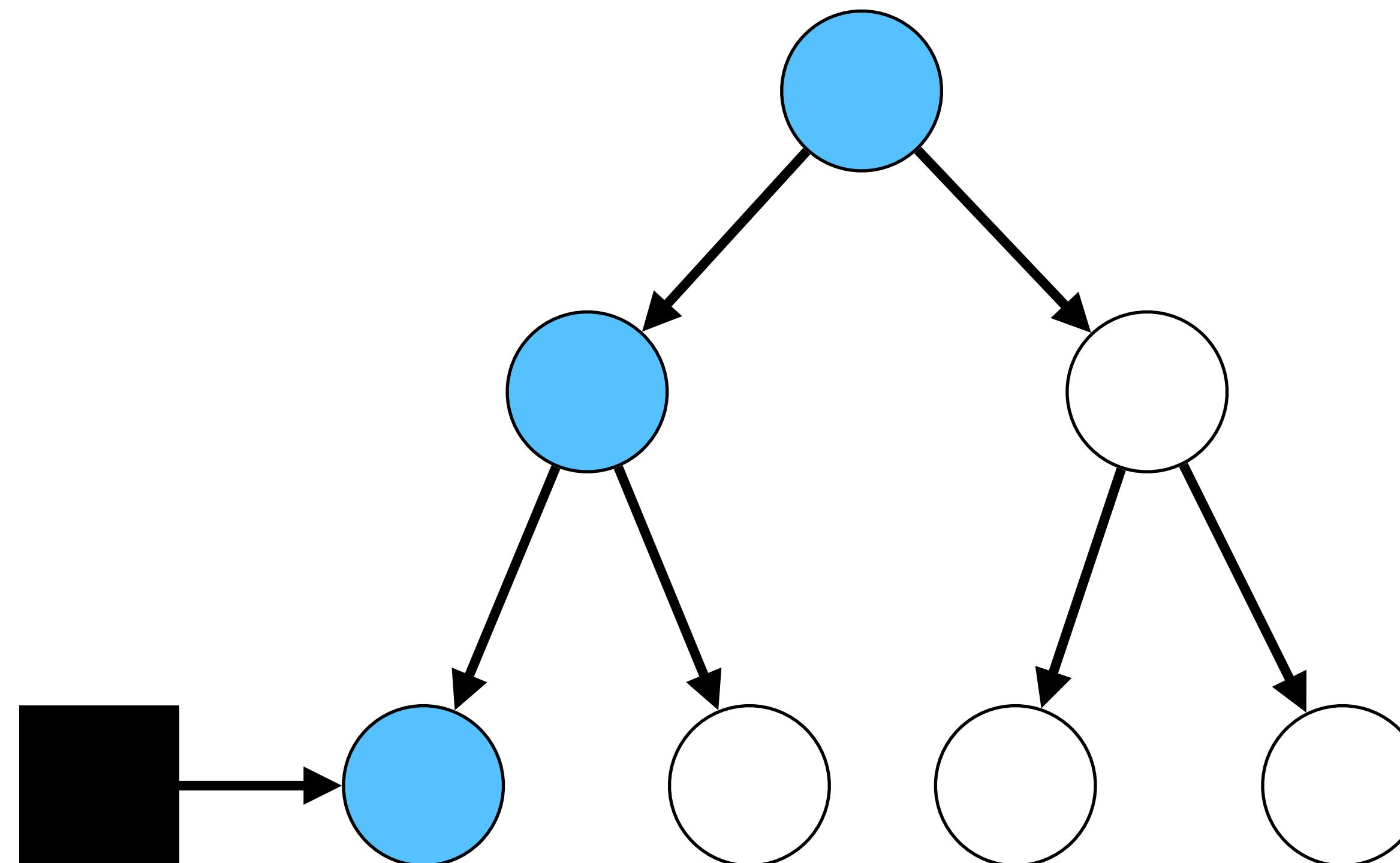
Original Categorization Approach

Greedy search and prediction from a single node



Original Categorization Approach

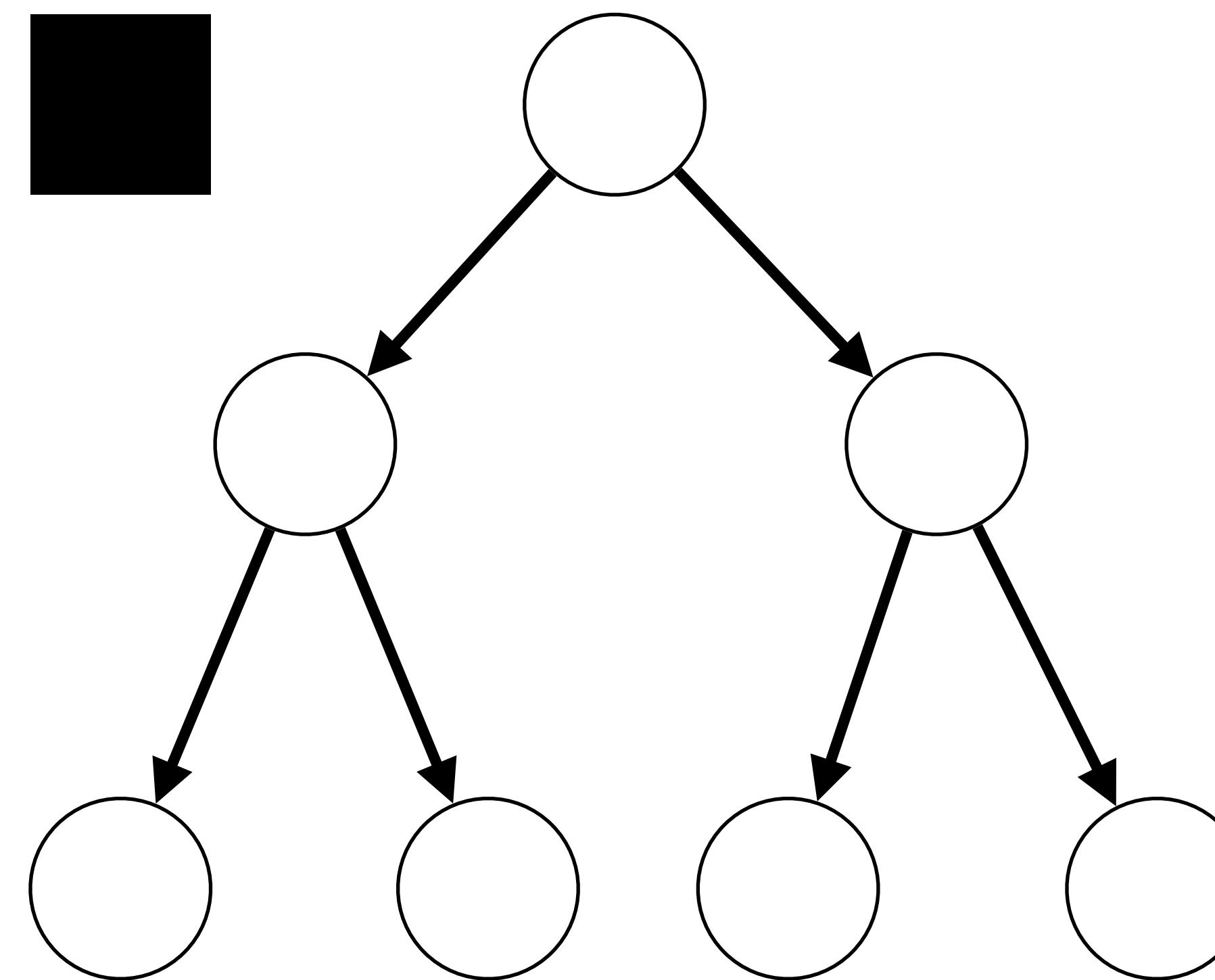
Greedy search and prediction from a single node



Output prediction about instance
using terminal node probability table

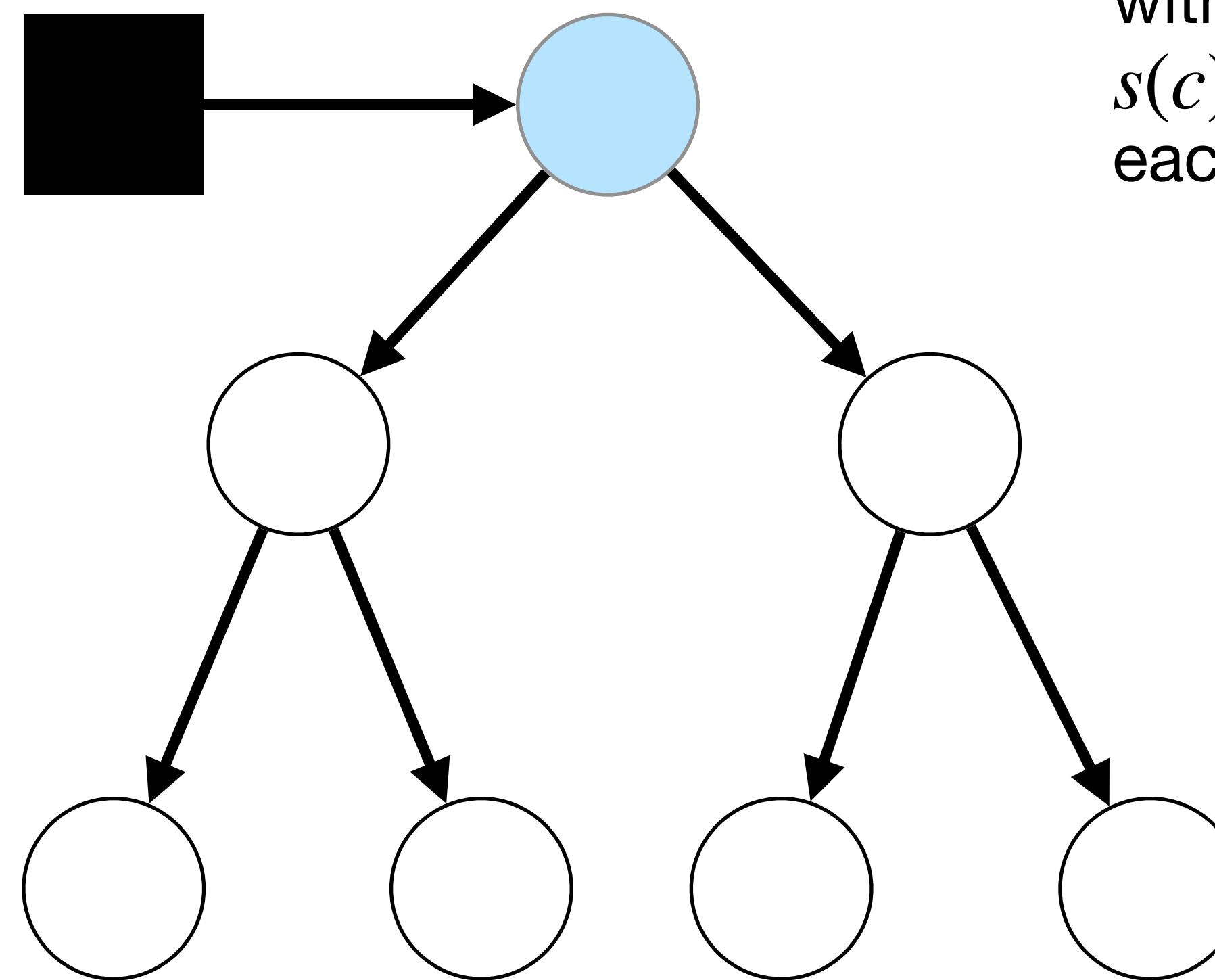
New Categorization Approach

Best first search and weighted prediction from all expanded nodes



New Categorization Approach

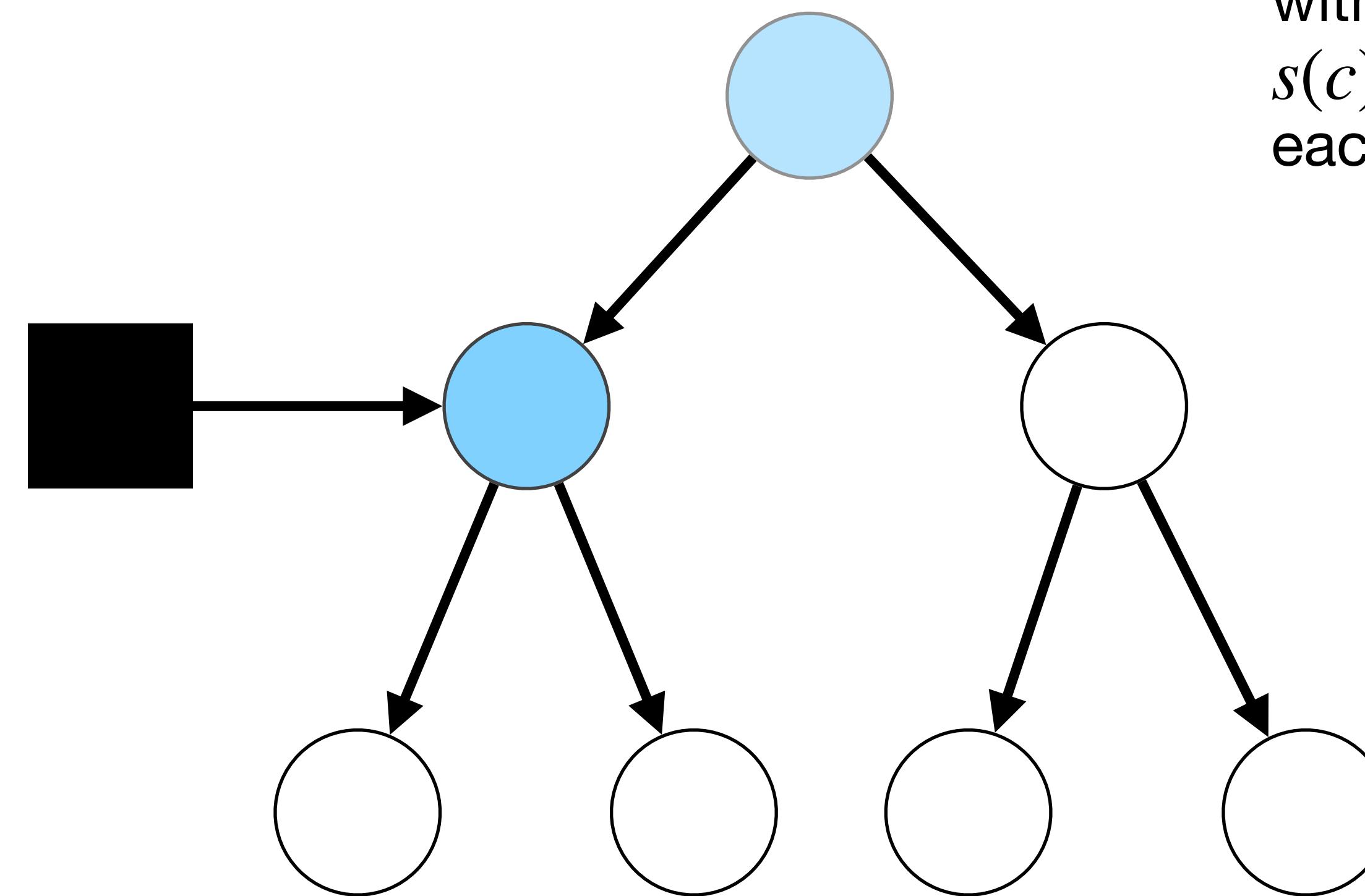
Best first search and weighted prediction from all expanded nodes



The node c on the search frontier with the highest collocation score $s(c) = P(x | c)P(c | x)$ expanded on each iteration.

New Categorization Approach

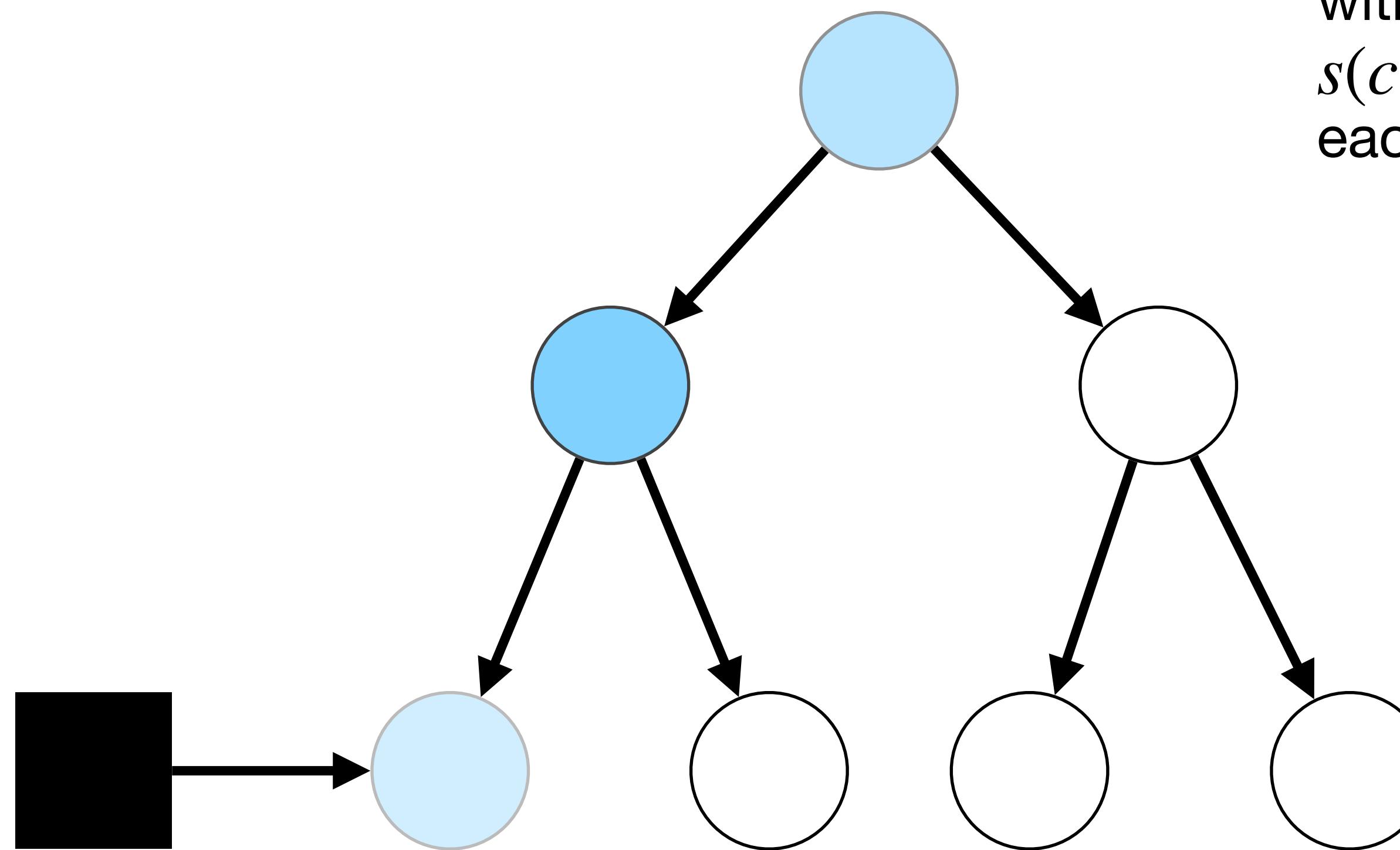
Best first search and weighted prediction from all expanded nodes



The node c on the search frontier with the highest collocation score $s(c) = P(x | c)P(c | x)$ expanded on each iteration.

New Categorization Approach

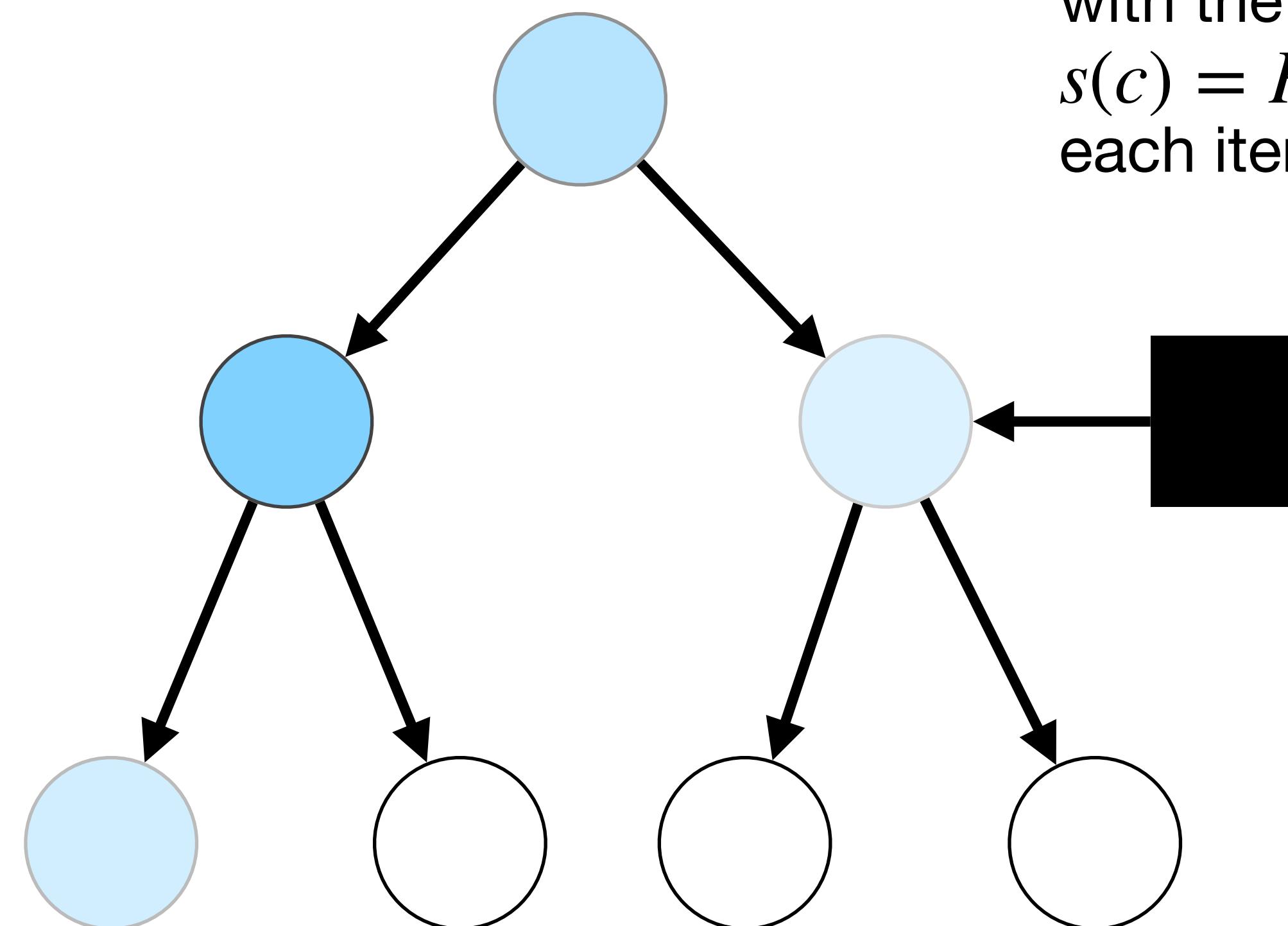
Best first search and weighted prediction from all expanded nodes



The node c on the search frontier with the highest collocation score $s(c) = P(x | c)P(c | x)$ expanded on each iteration.

New Categorization Approach

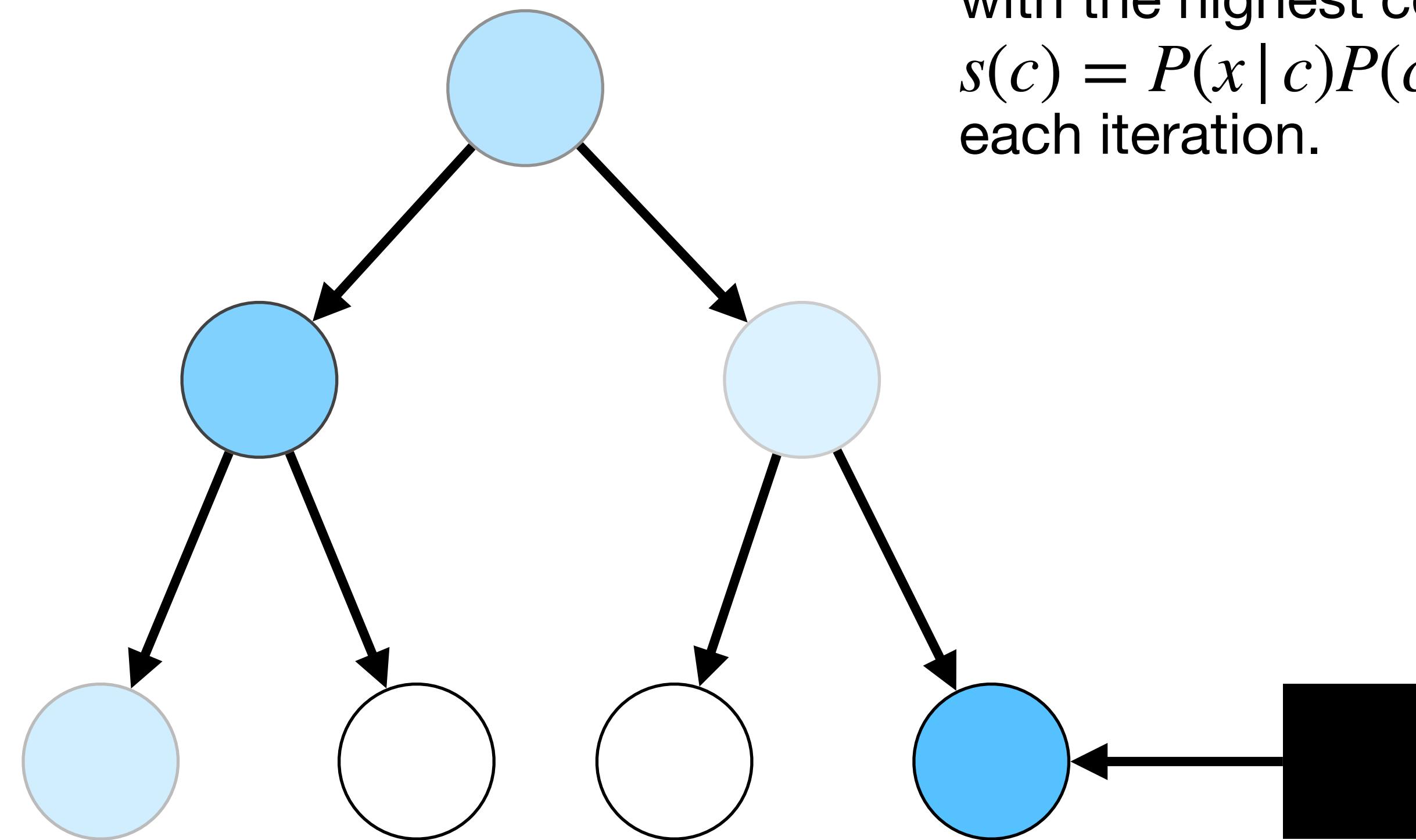
Best first search and weighted prediction from all expanded nodes



The node c on the search frontier with the highest collocation score $s(c) = P(x | c)P(c | x)$ expanded on each iteration.

New Categorization Approach

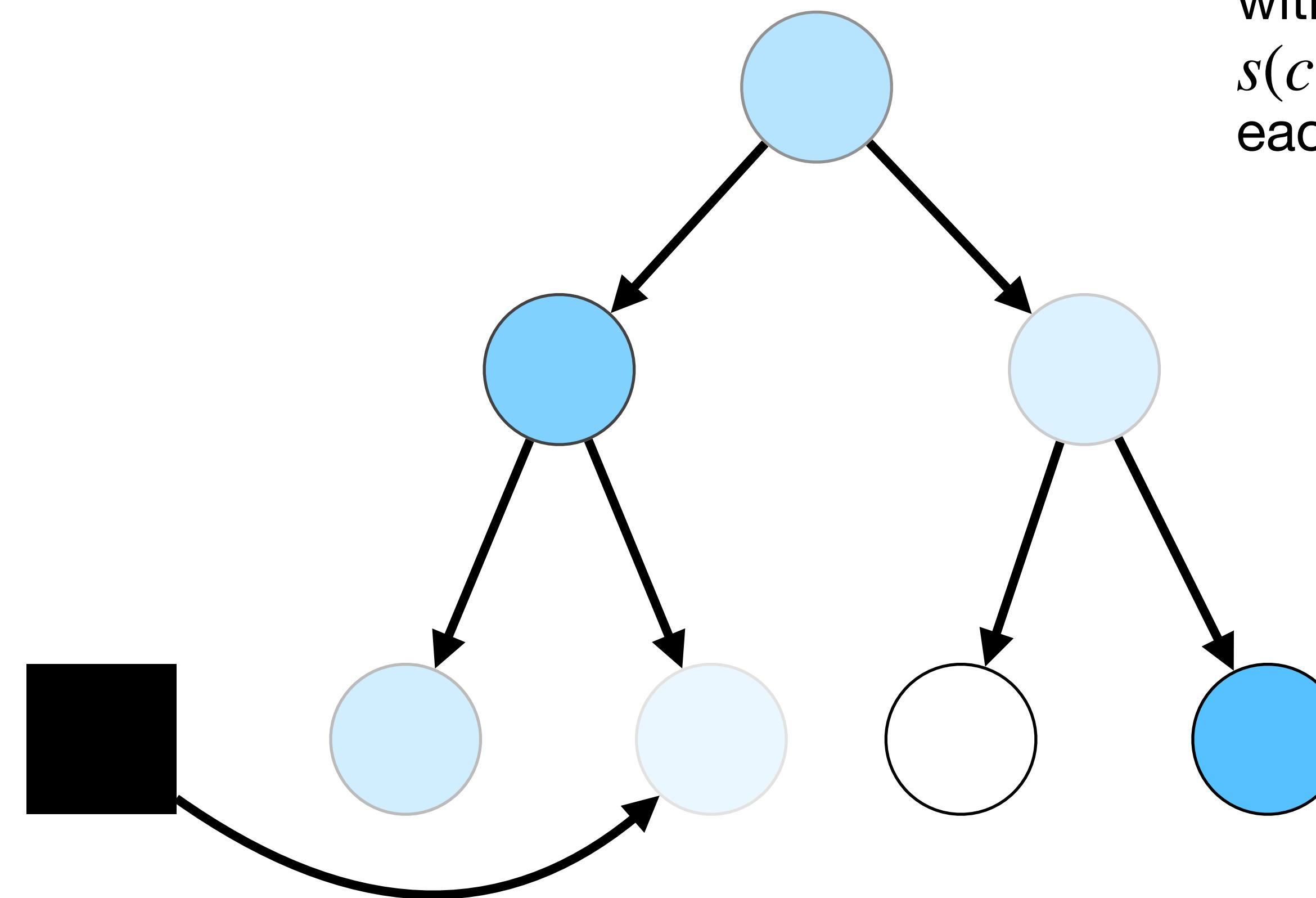
Best first search and weighted prediction from all expanded nodes



The node c on the search frontier with the highest collocation score $s(c) = P(x | c)P(c | x)$ expanded on each iteration.

New Categorization Approach

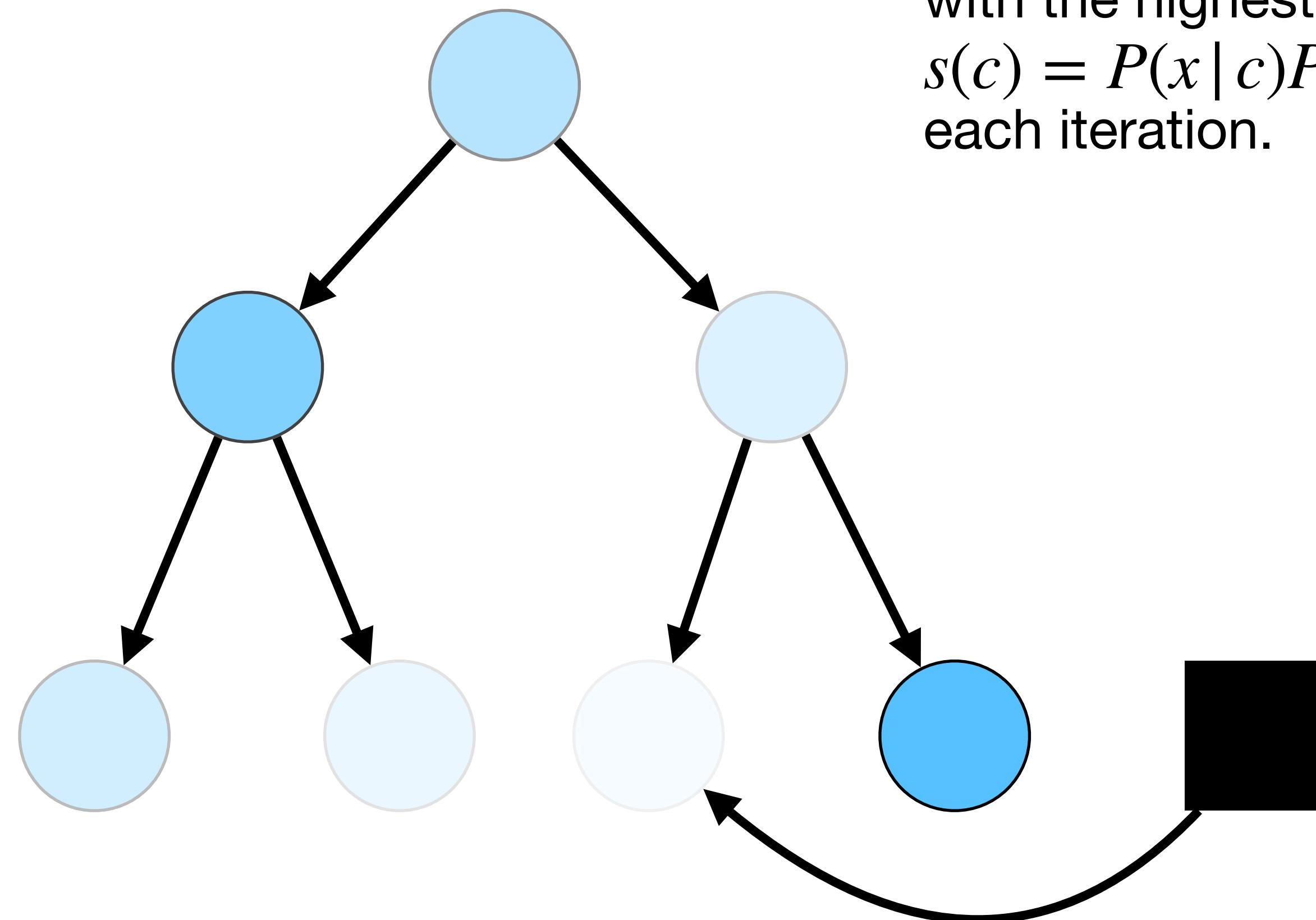
Best first search and weighted prediction from all expanded nodes



The node c on the search frontier with the highest collocation score $s(c) = P(x | c)P(c | x)$ expanded on each iteration.

New Categorization Approach

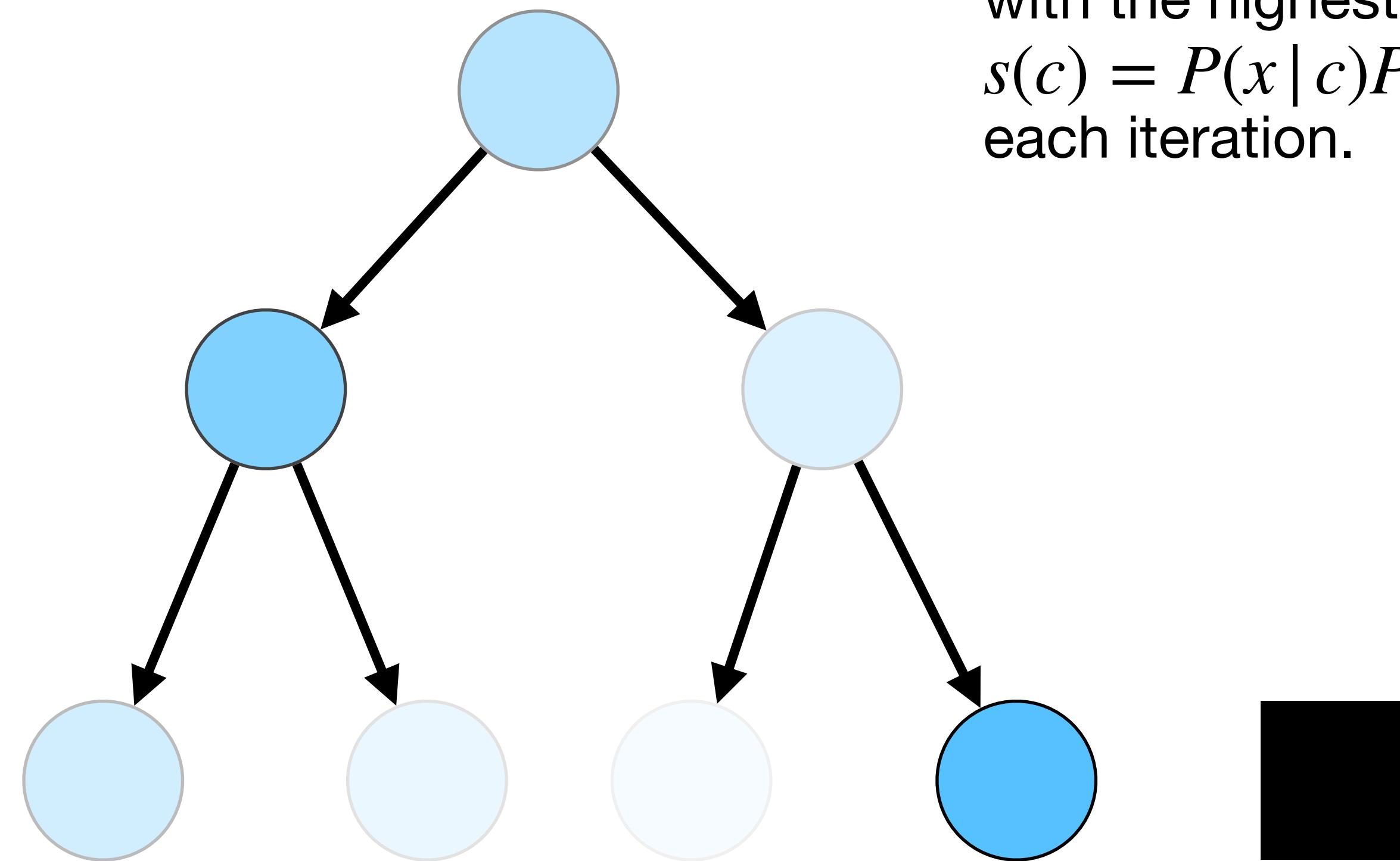
Best first search and weighted prediction from all expanded nodes



The node c on the search frontier with the highest collocation score $s(c) = P(x | c)P(c | x)$ expanded on each iteration.

New Categorization Approach

Best first search and weighted prediction from all expanded nodes

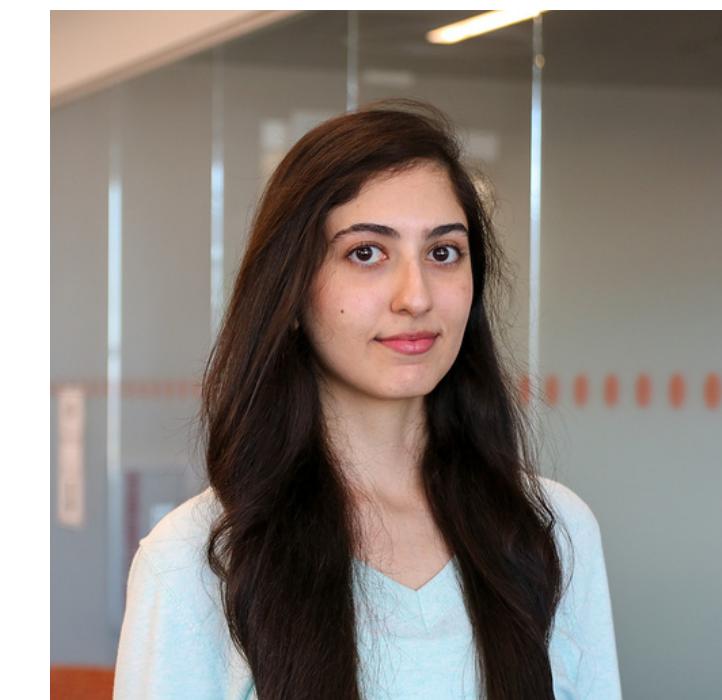


The node c on the search frontier with the highest collocation score $s(c) = P(x | c)P(c | x)$ expanded on each iteration.



Cobweb's final prediction is the combination of predictions from all expanded nodes, weighted by their collocation.

Cobweb/4V: Incremental learning over image data



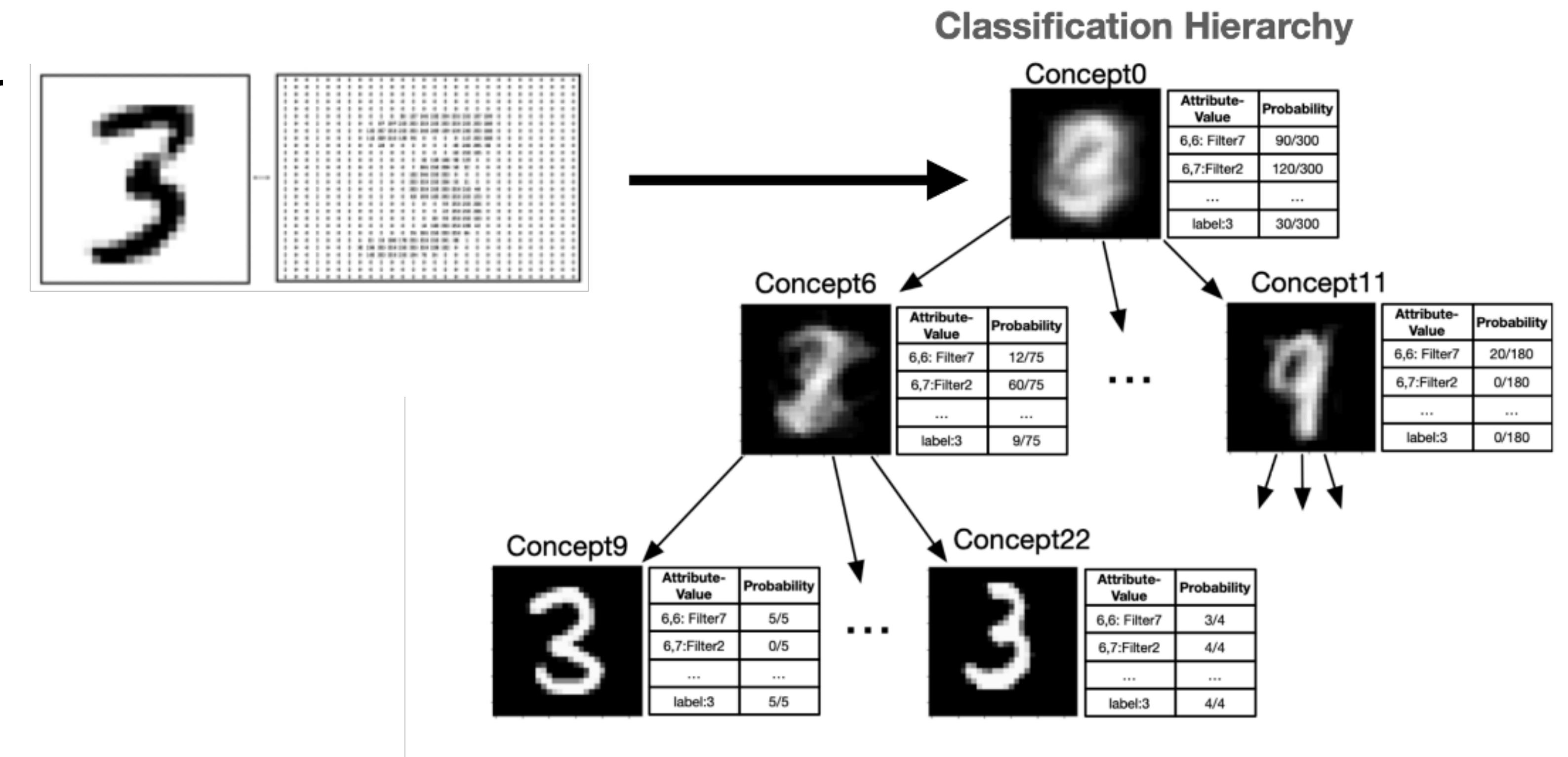
Nicki Barari
nb895@drexel.edu



Xin Lian
xlian34@gatech.edu

Tensor Representation

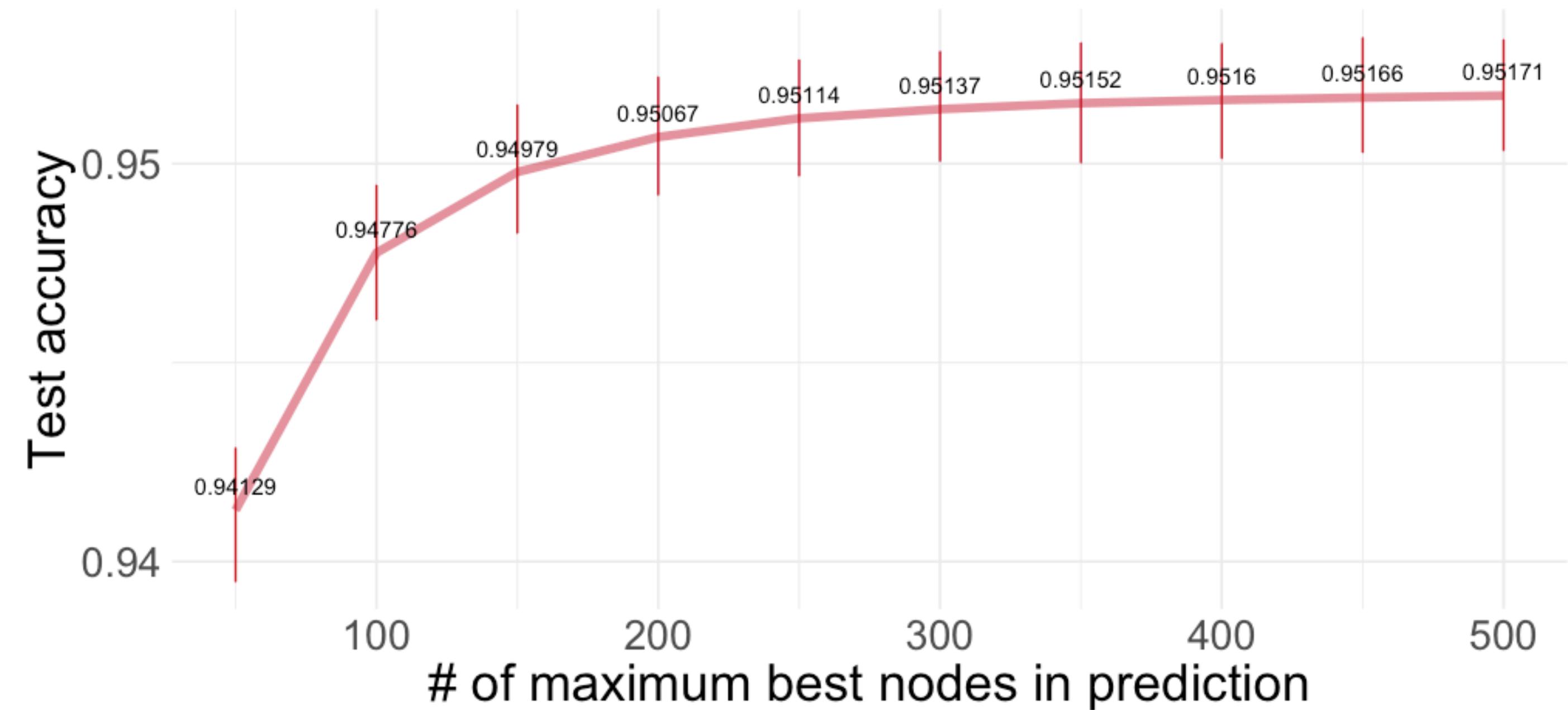
- Cobweb/4V uses a tensor representation, where each image is a tensor of pixel channel intensities.
- Building on Cobweb/3, it stores statistics in each node to efficiently compute and update means and variances online without needing to iterate over prior data.



Evaluation of Cobweb Changes on MNIST

More nodes expanded yields better performance

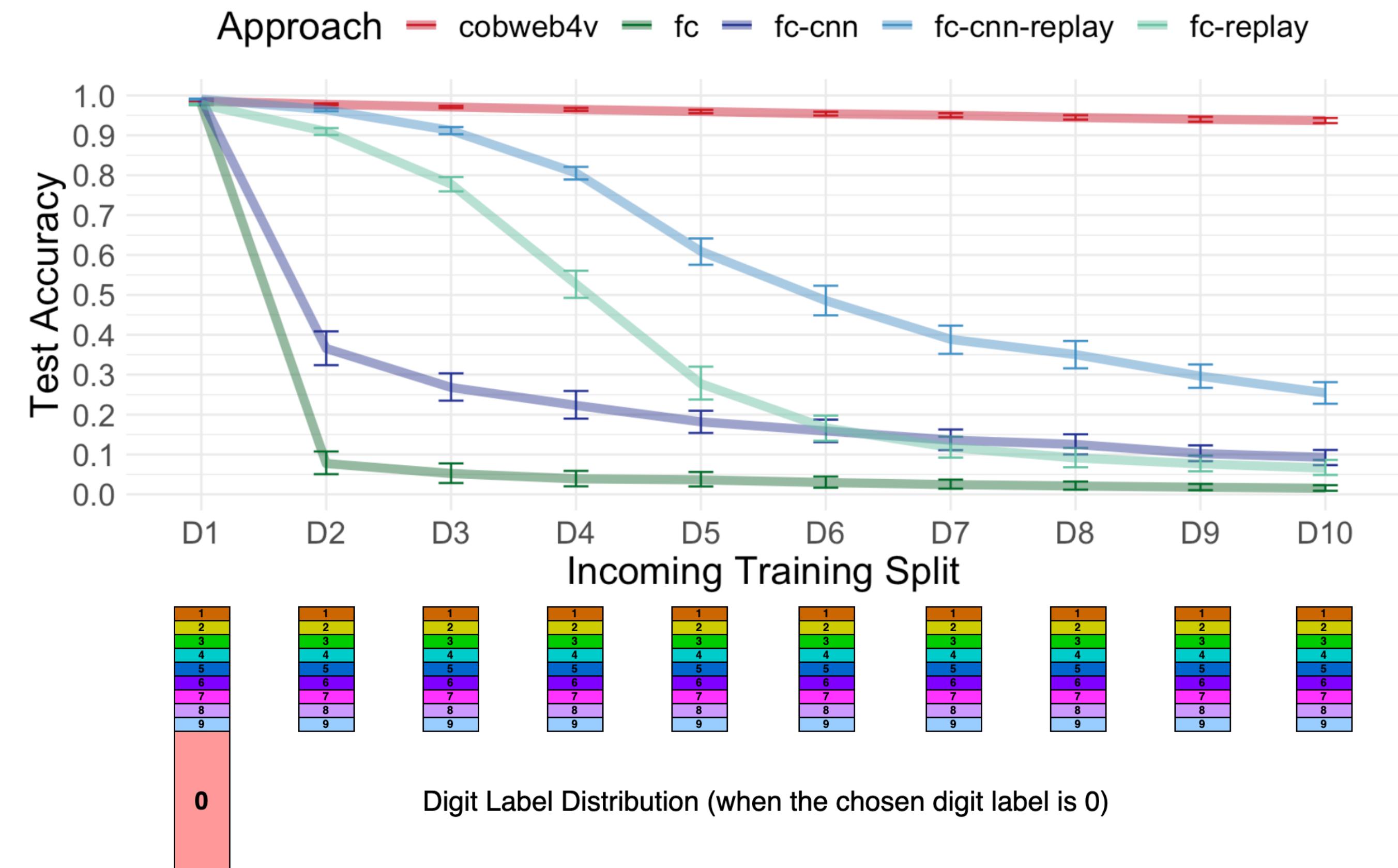
- We trained on all the MNIST training data (60k images) and evaluated on the official test set (10k images).
- We varied the maximum number of nodes expanded during prediction from 50 - 500 in increments of 50.
- We found that in general, the more nodes expanded the better, but that performance levels off around 300 nodes expanded.



An Evaluation of Catastrophic Forgetting

Cobweb/4V Does Not Catastrophically Forget

- We compared Cobweb/4V to the FC and FC-CNN on an class-incremental prediction
- It sees all of a target digit up front (0 in example), then every successive split of training data lacks the digit
- We evaluate on all test items for the target digit
- We also compared with variants of FC and FC-CNN that use a replay buffer
- Our results show that all NN approaches forget catastrophically, but Cobweb/4V does not



An Evaluation of Learning & Performance

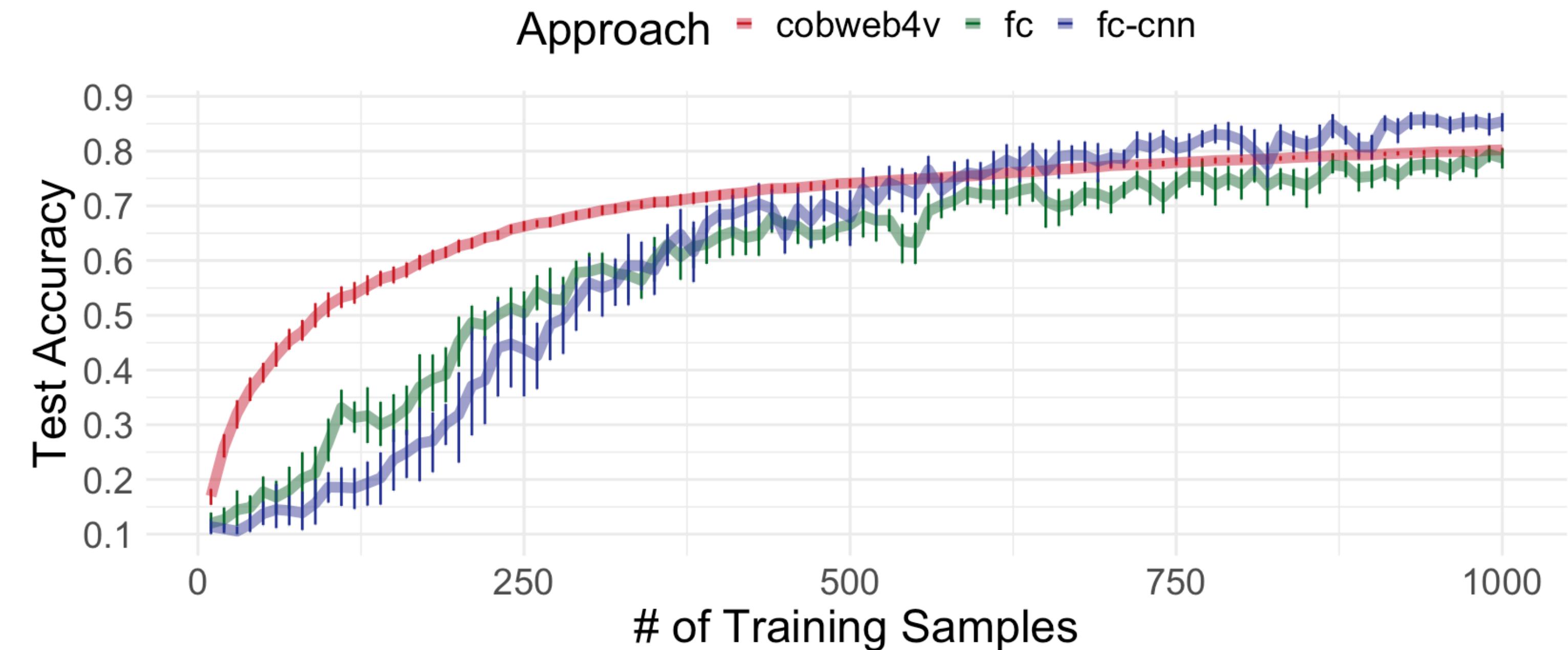
Cobweb exhibits fast, stable learning

We compared Cobweb/4V to two neural network baselines on an incremental MNIST training task.

We presented each approach 10 images at a time and then evaluated it on entire test set.

Cobweb has fast, stable learning, performing much better in cases with fewer examples.

We also compared Cobweb to the NN baselines after training on entire MNIST training set (see table) and found it is comparable to FC.



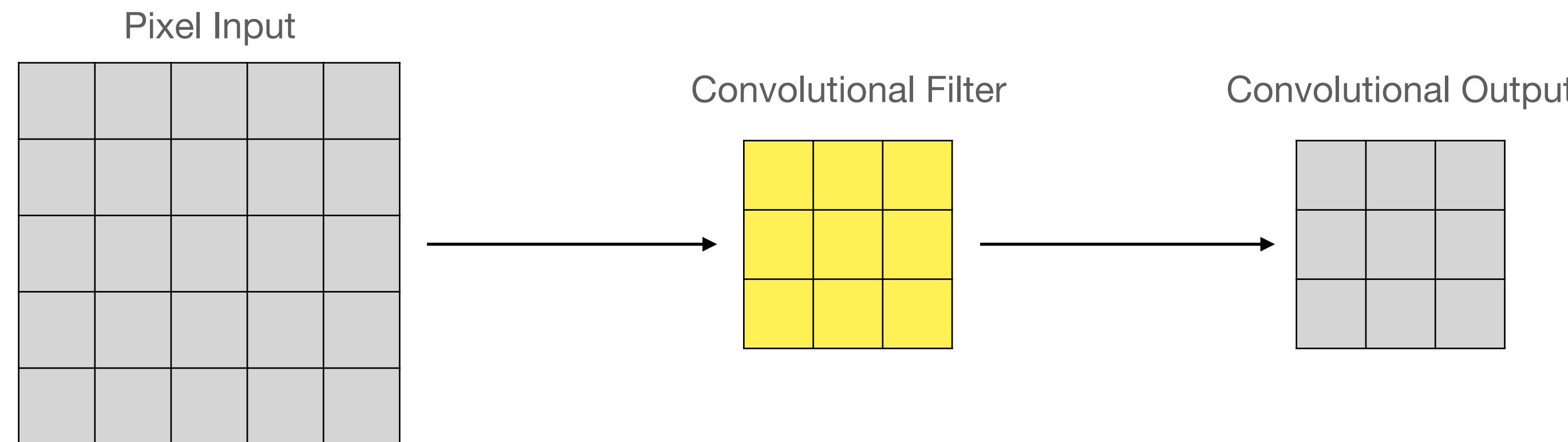
FC is a standard fully-connected neural network (with 1 hidden layer) and FC-CNN is extended with 2 convolutional and max pool layers.

	Final Accuracy (after all 60k training)
Cobweb/4V	95.14%
FC	95.13%
FC-CNN	97.35%

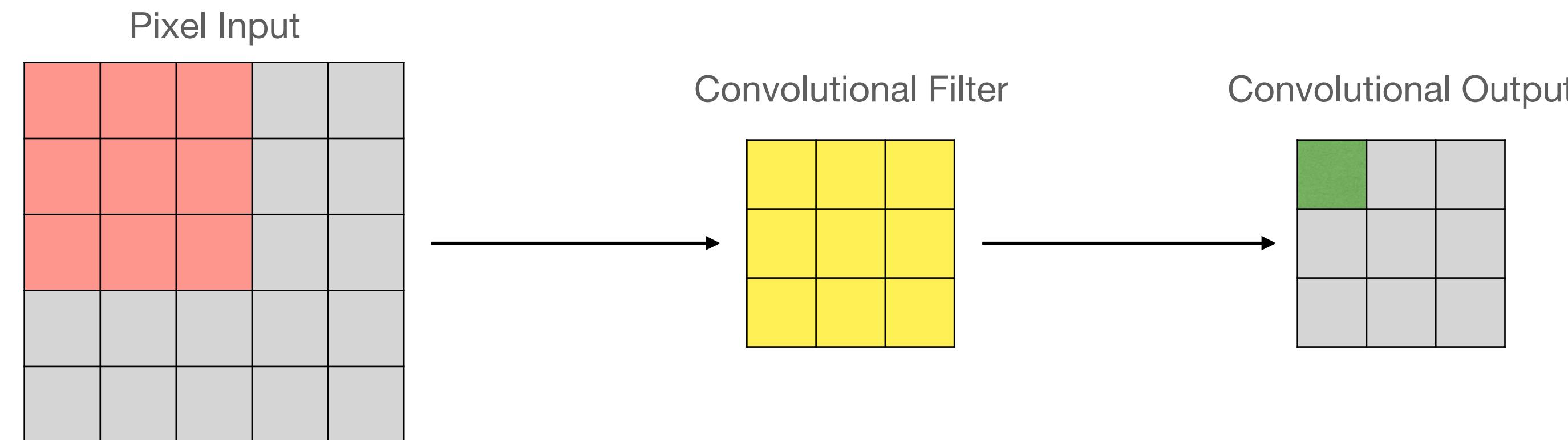
Convolutional Cobweb

MacLellan., C.J. & Thakur, H. (2021). Convolutional Cobweb: A Model of Incremental Learning from 2D Images. In *Proceedings of the Ninth Annual Conference on Advances in Cognitive Systems*. ([pdf](#)) ([talk](#))

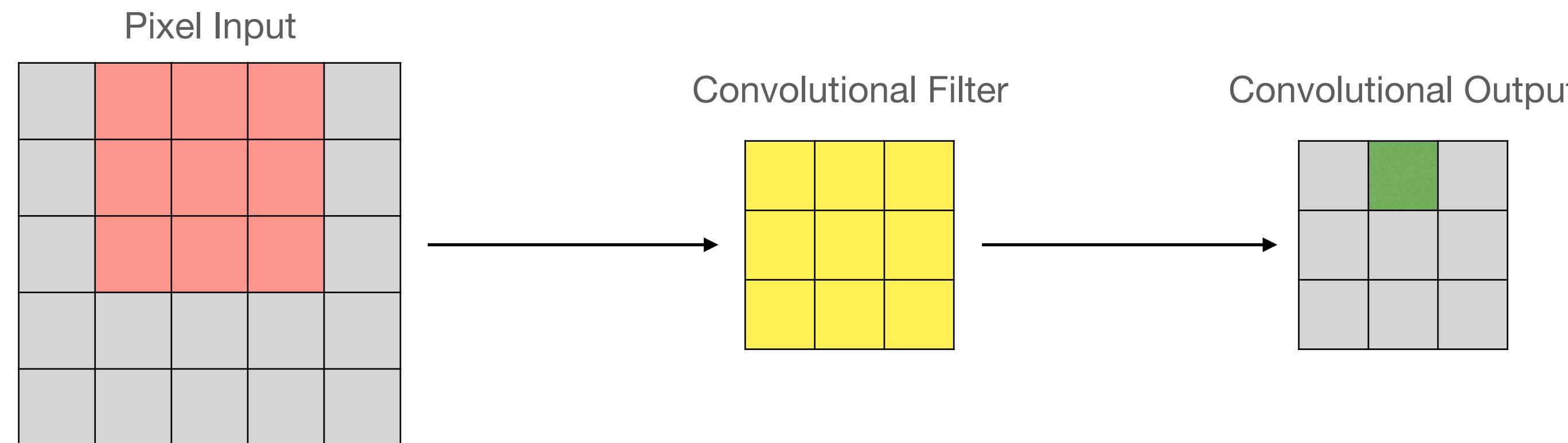
Convolutional Processing in a Simple CNN



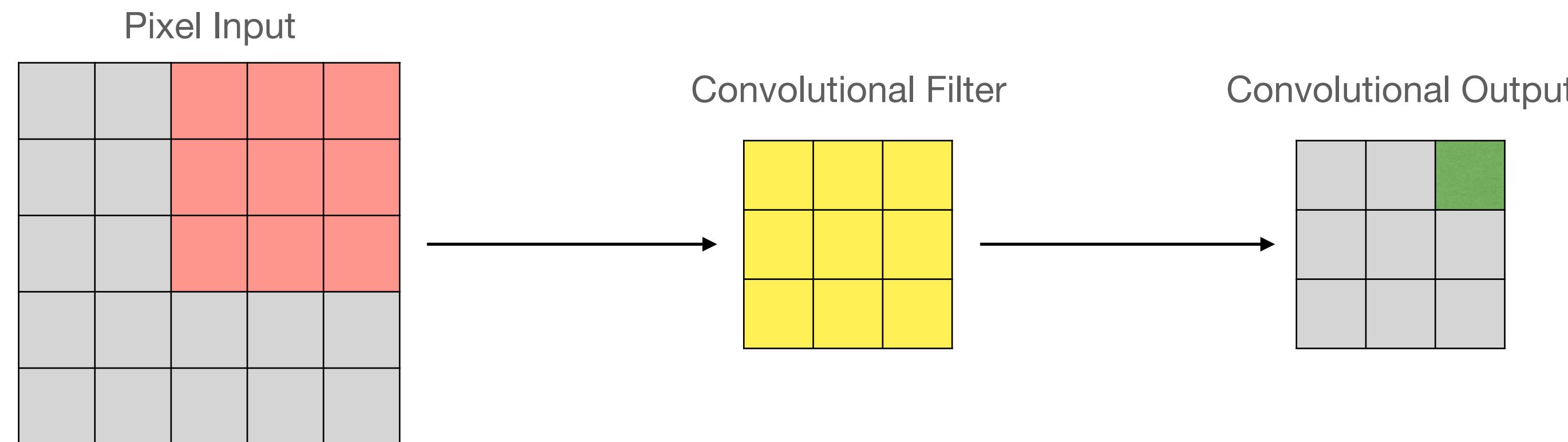
Convolutional Processing in a Simple CNN



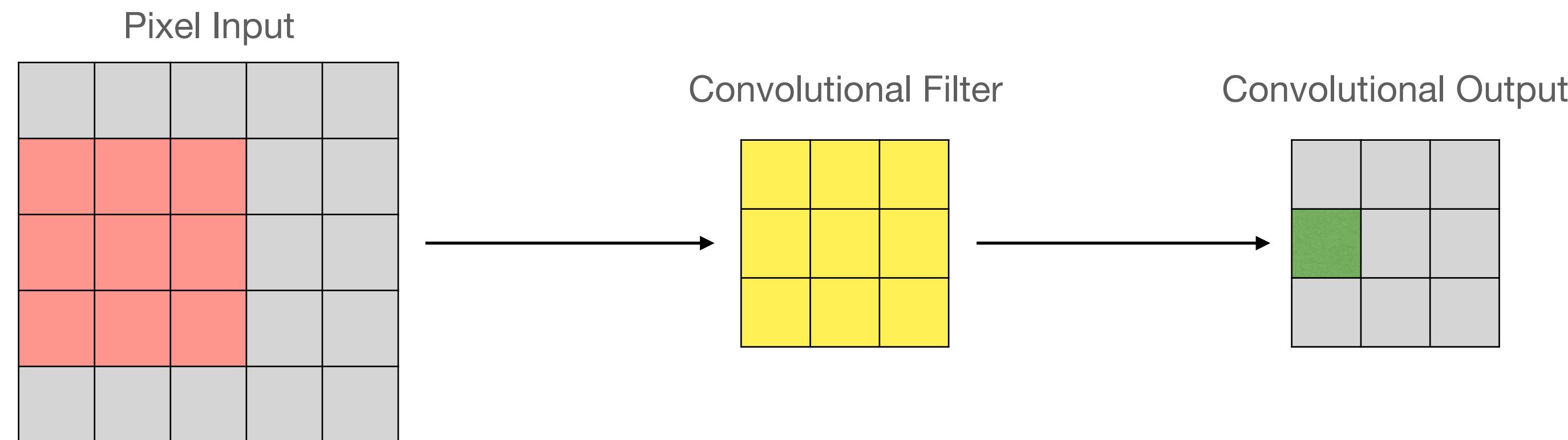
Convolutional Processing in a Simple CNN



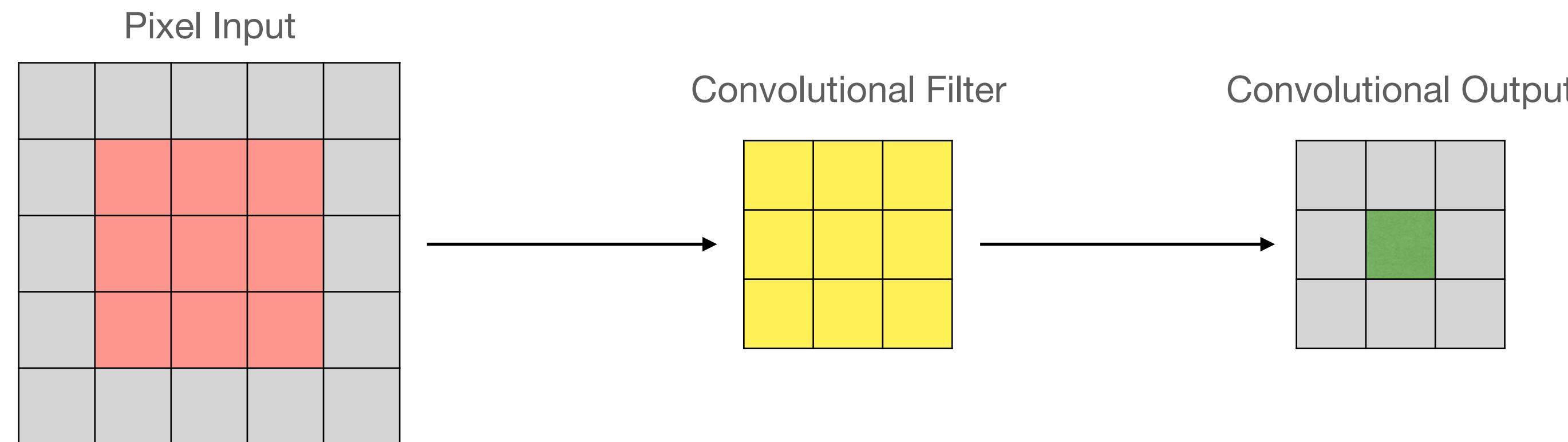
Convolutional Processing in a Simple CNN



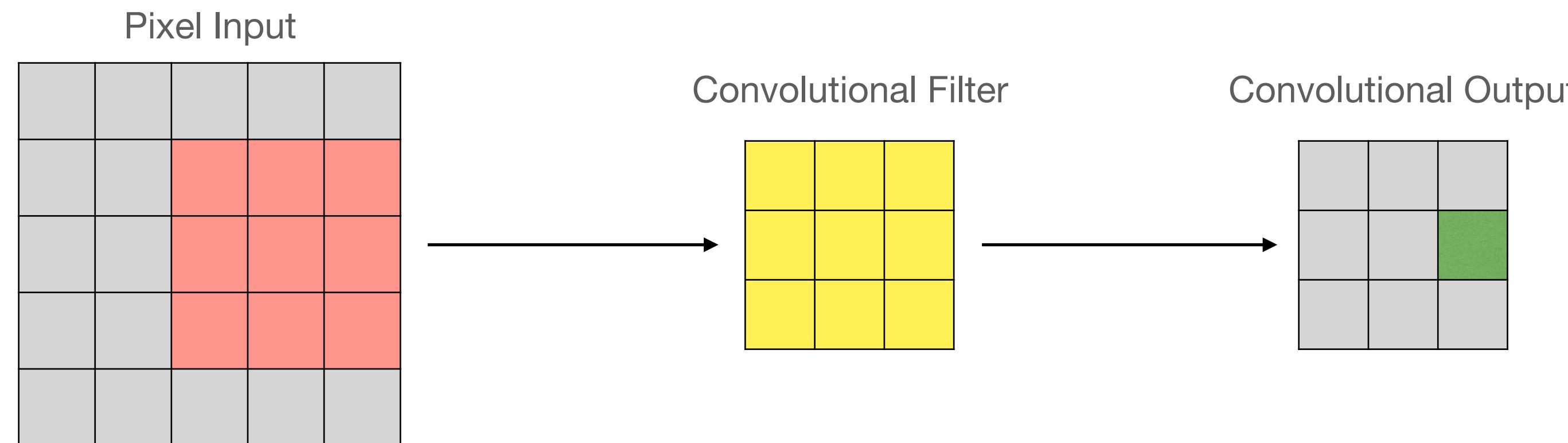
Convolutional Processing in a Simple CNN



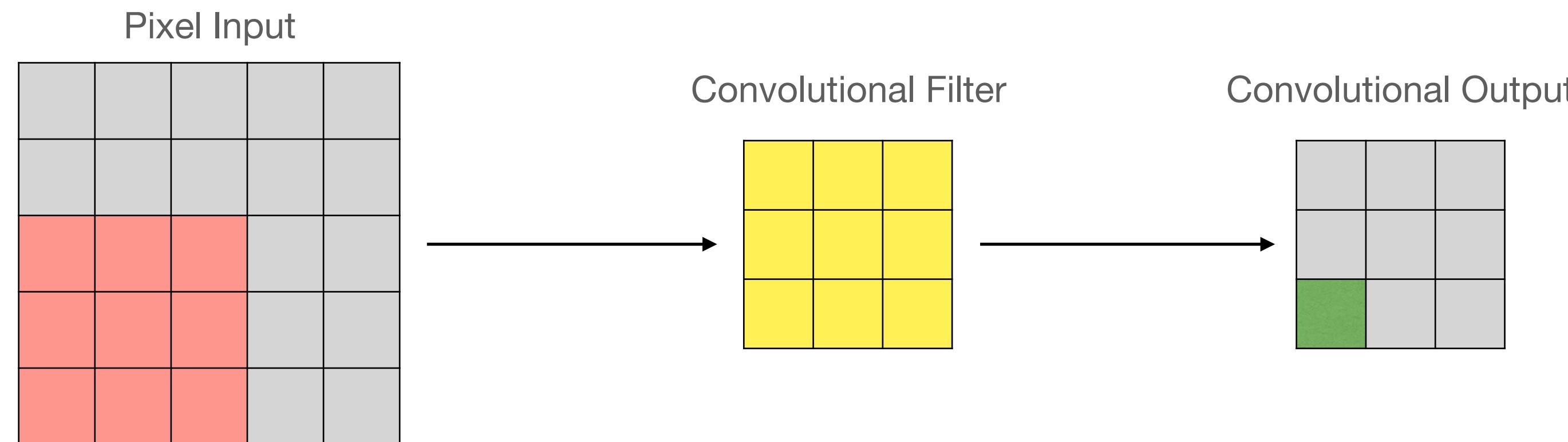
Convolutional Processing in a Simple CNN



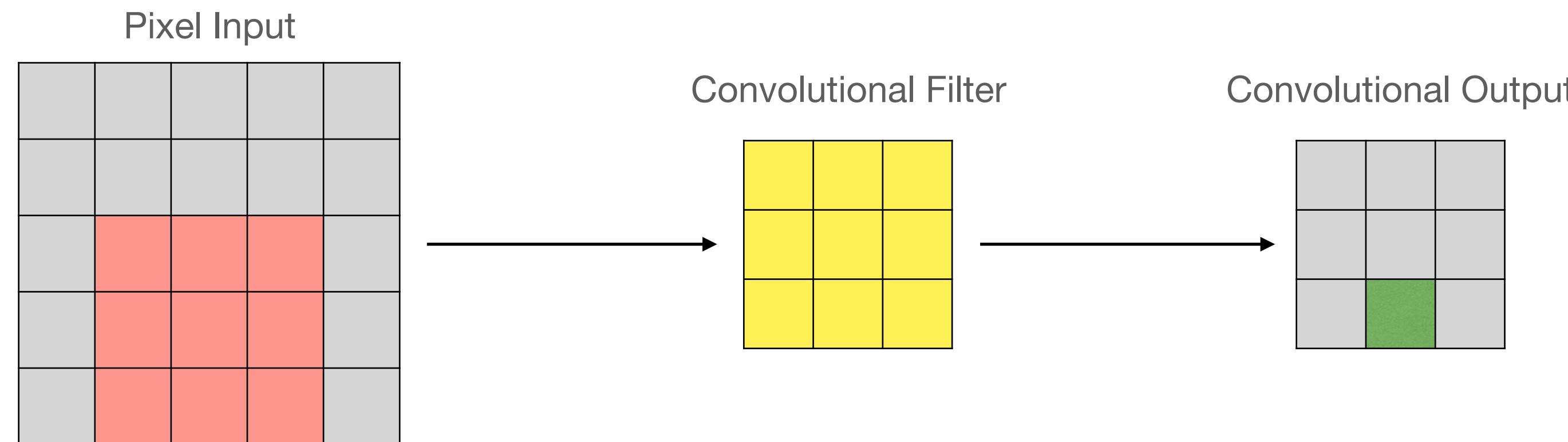
Convolutional Processing in a Simple CNN



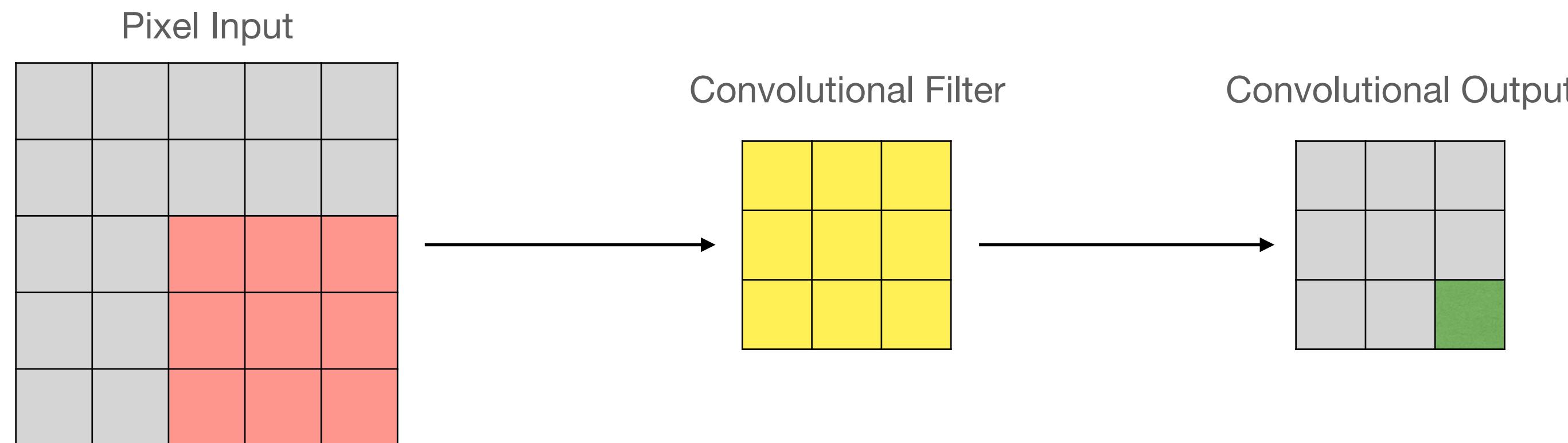
Convolutional Processing in a Simple CNN



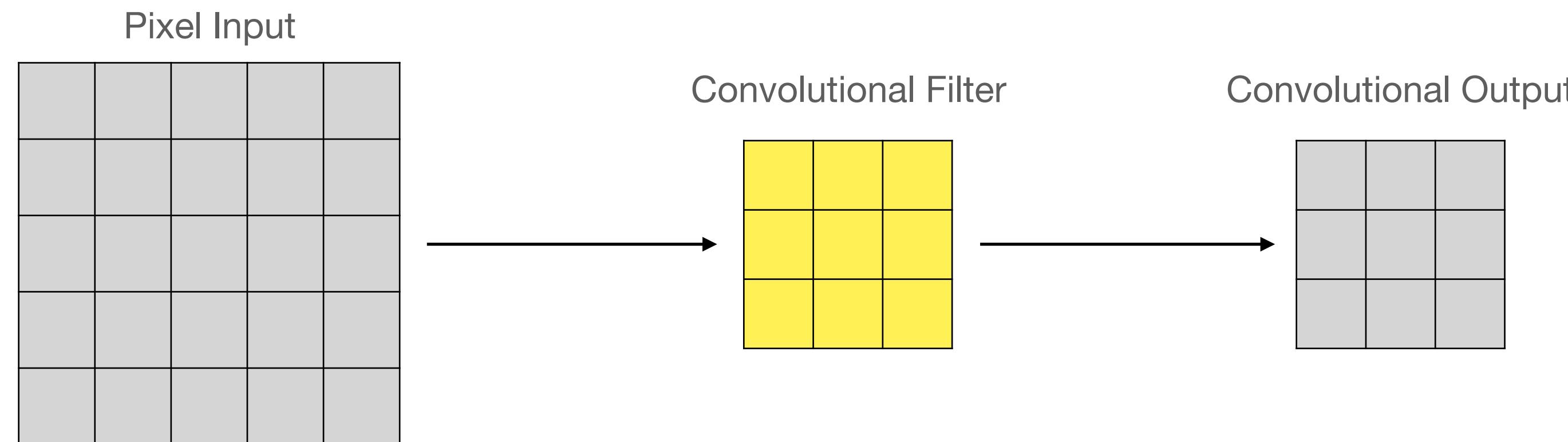
Convolutional Processing in a Simple CNN



Convolutional Processing in a Simple CNN

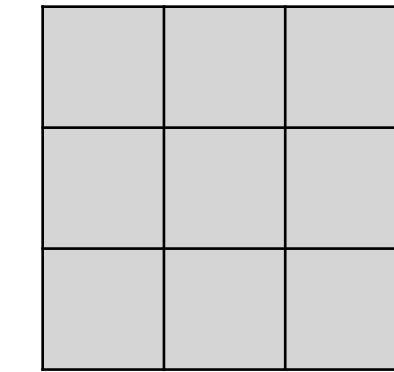


Convolutional Processing in a Simple CNN



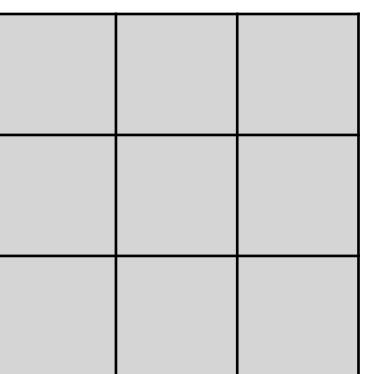
Convolutional Processing in a Simple CNN

Convolutional Output

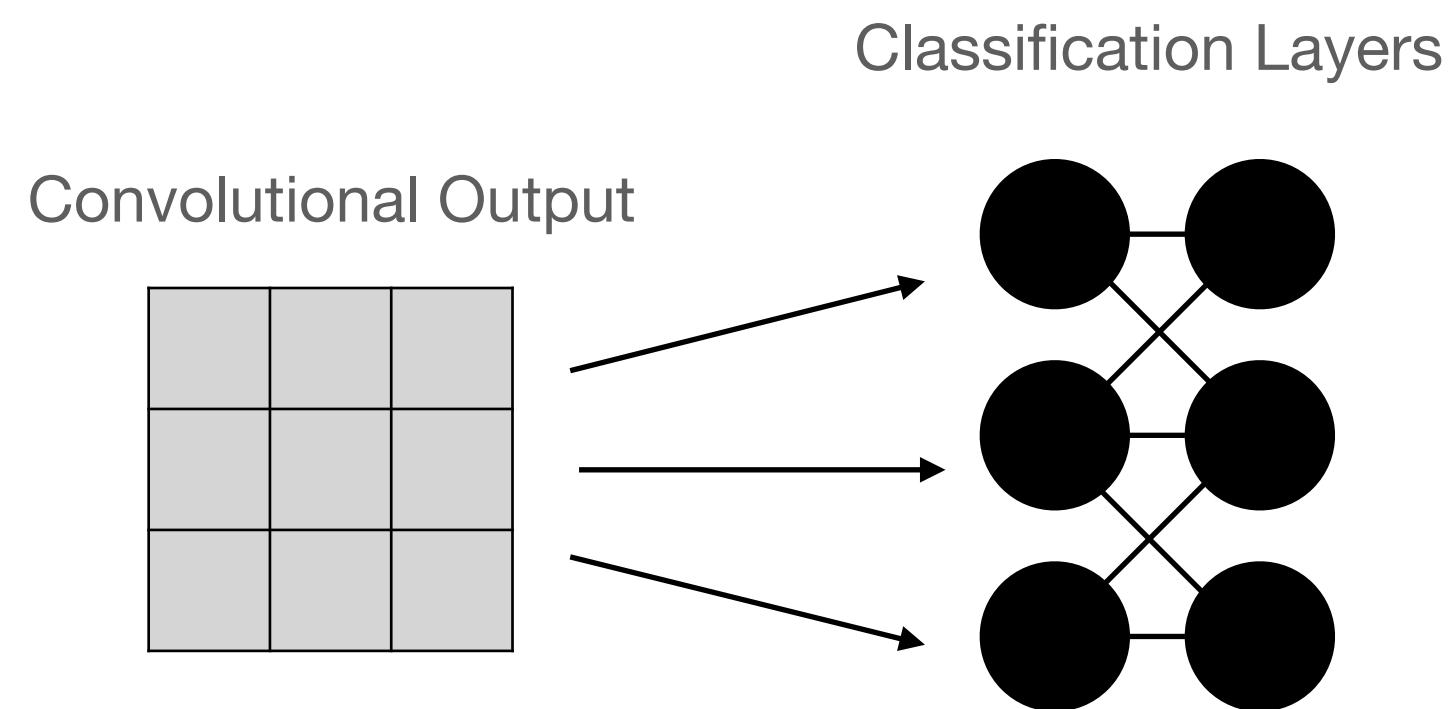


Convolutional Processing in a Simple CNN

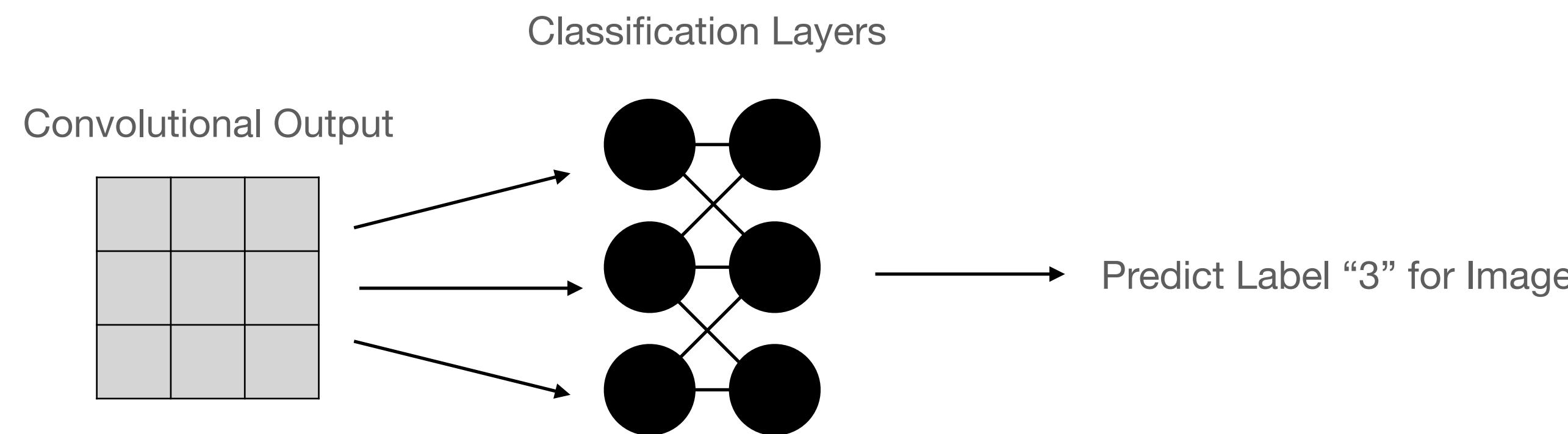
Convolutional Output



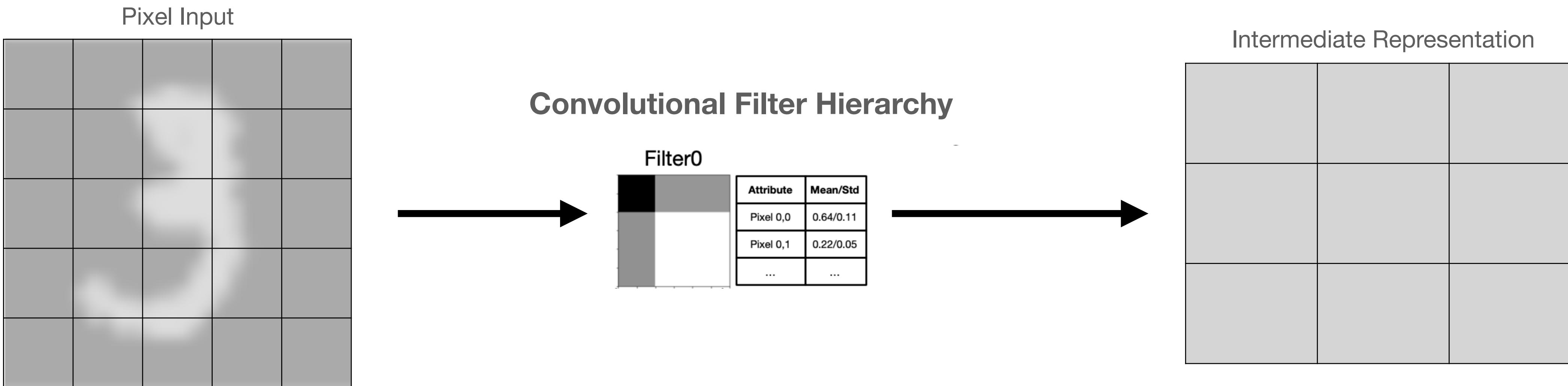
Convolutional Processing in a Simple CNN



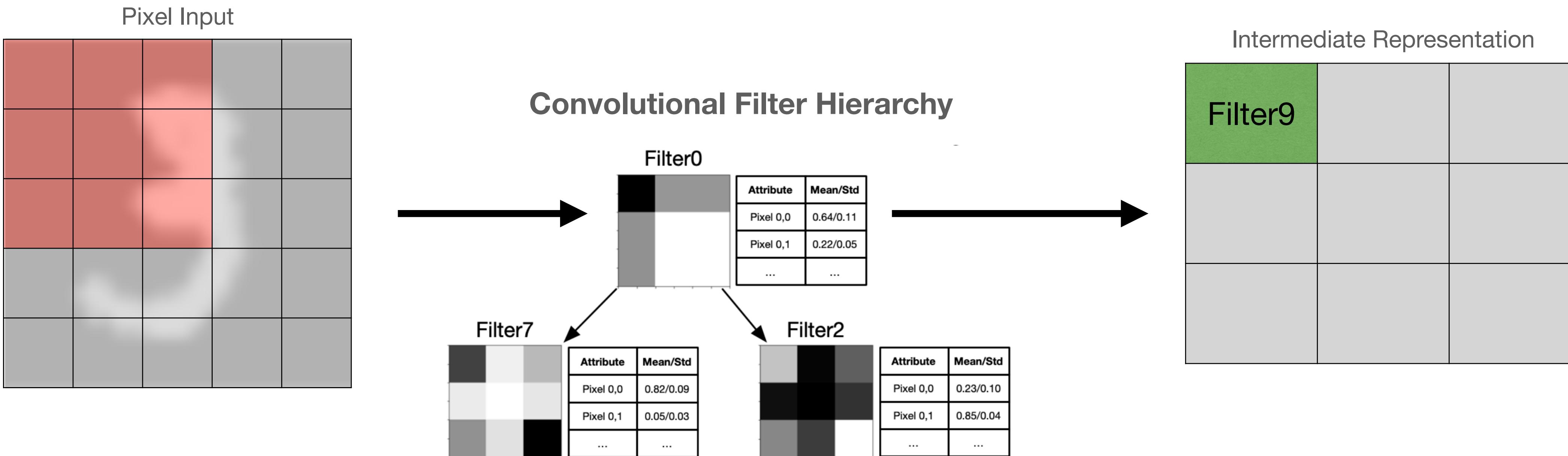
Convolutional Processing in a Simple CNN



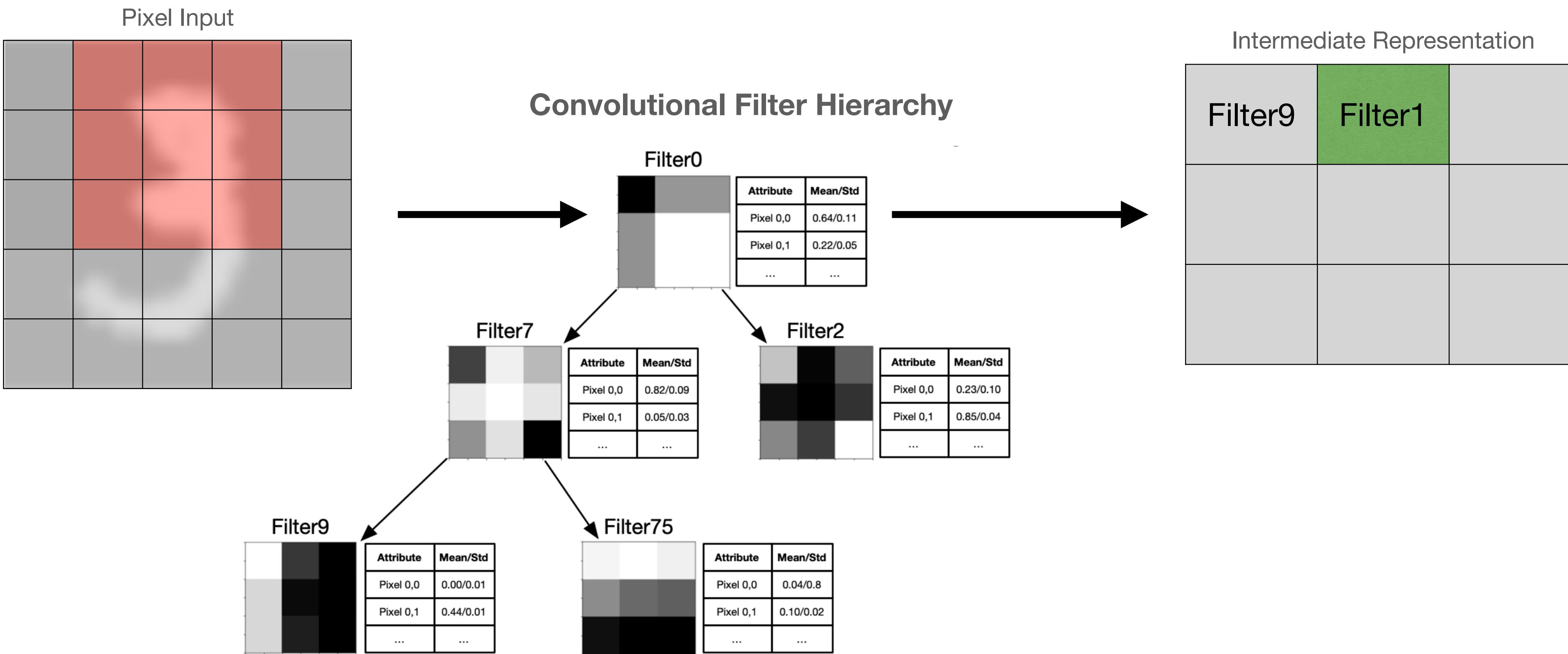
Convolutional Cobweb Approach



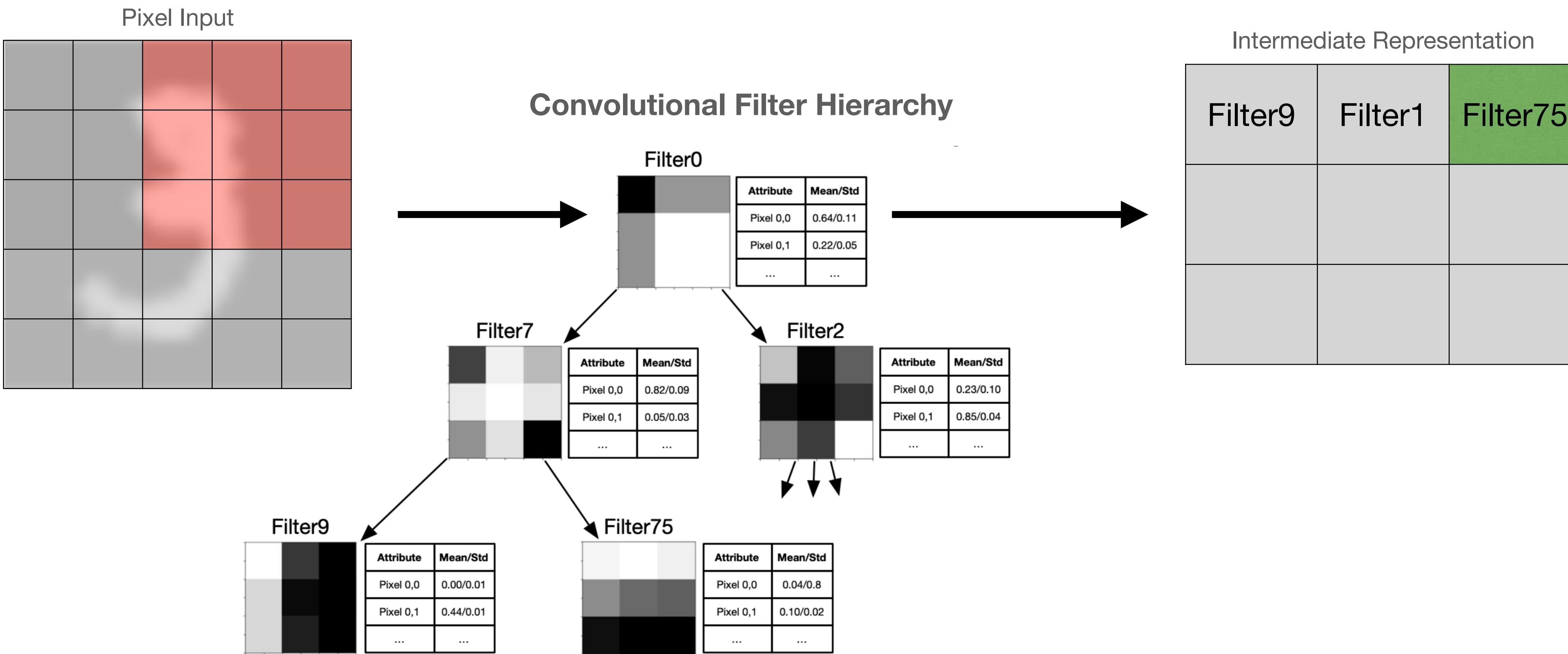
Convolutional Cobweb Approach



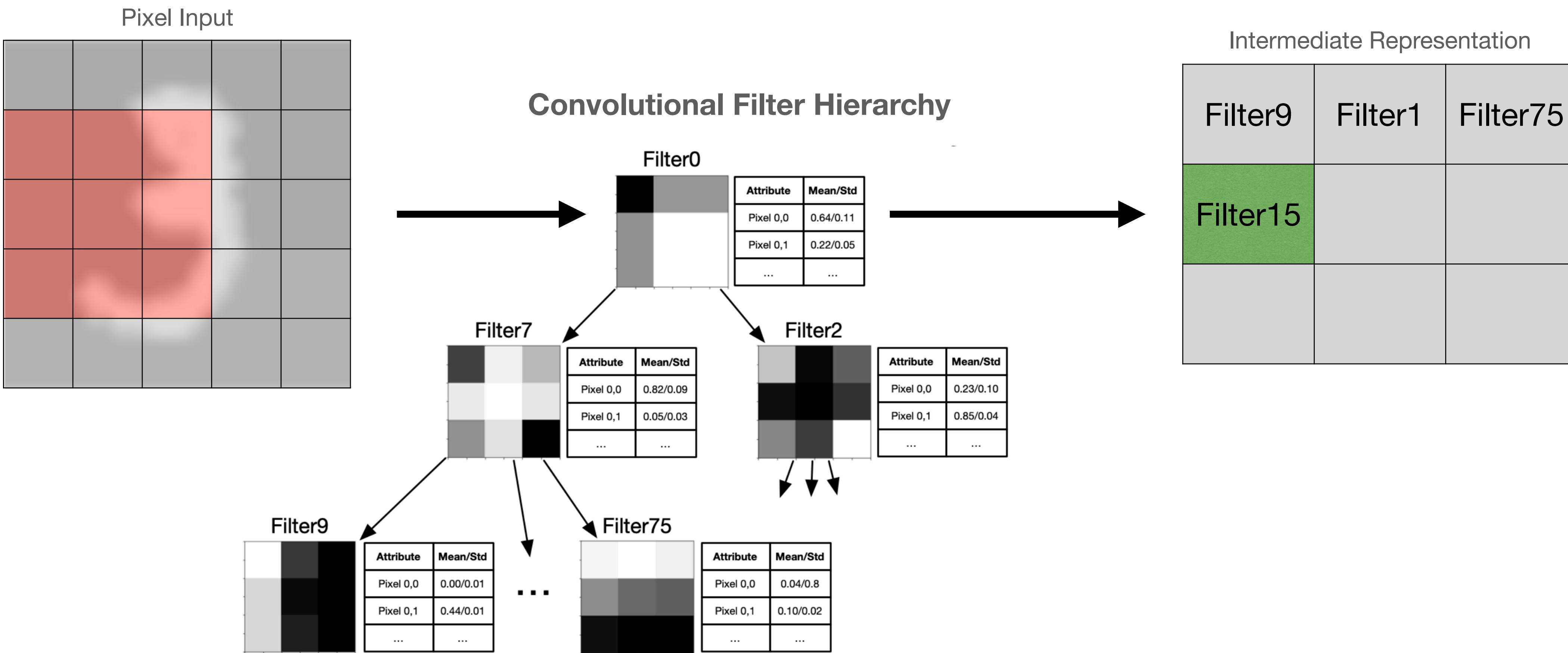
Convolutional Cobweb Approach



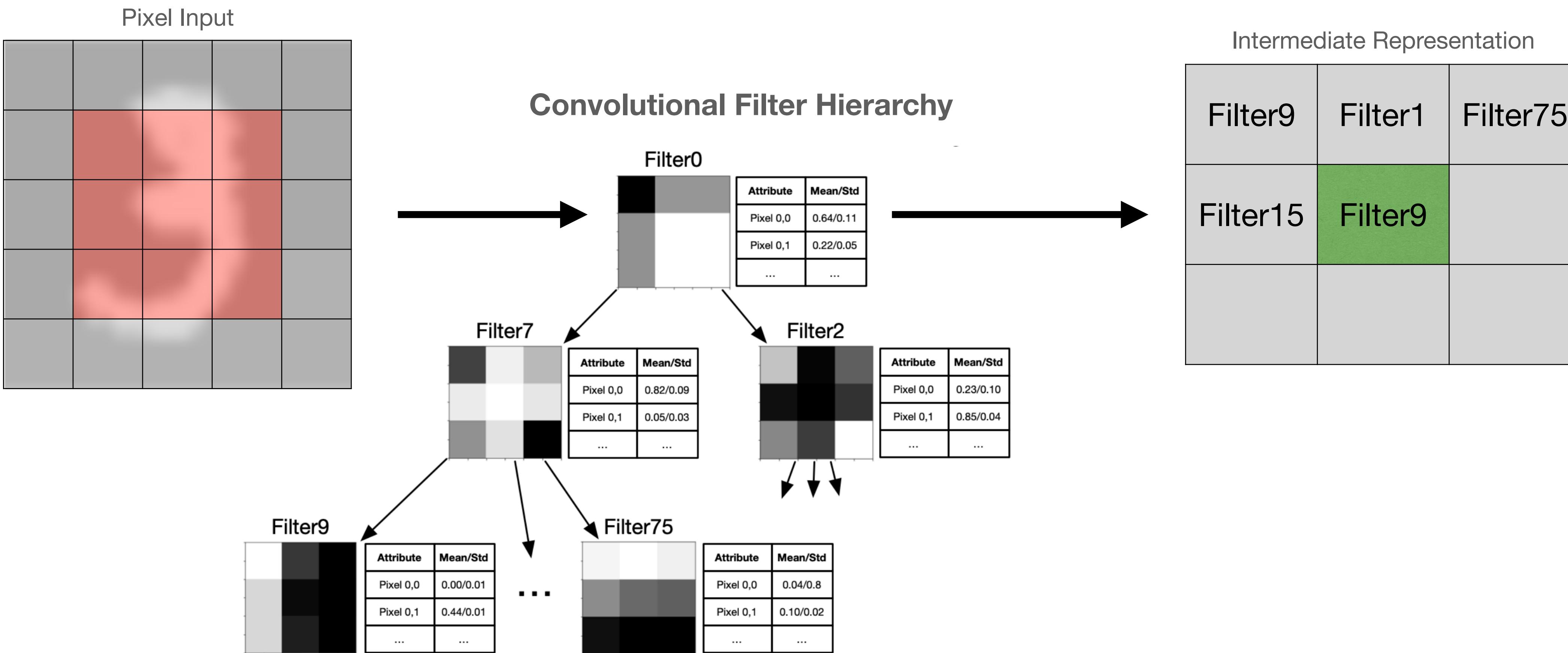
Convolutional Cobweb Approach



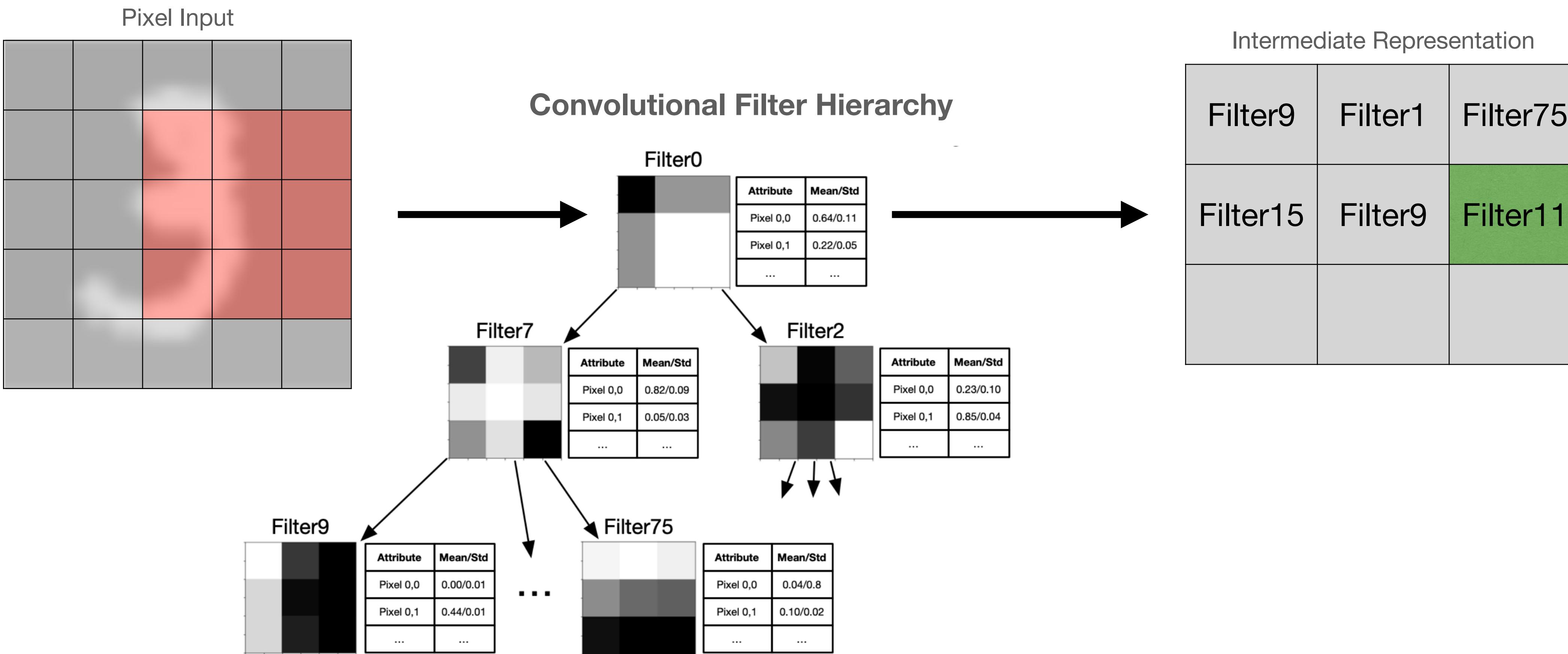
Convolutional Cobweb Approach



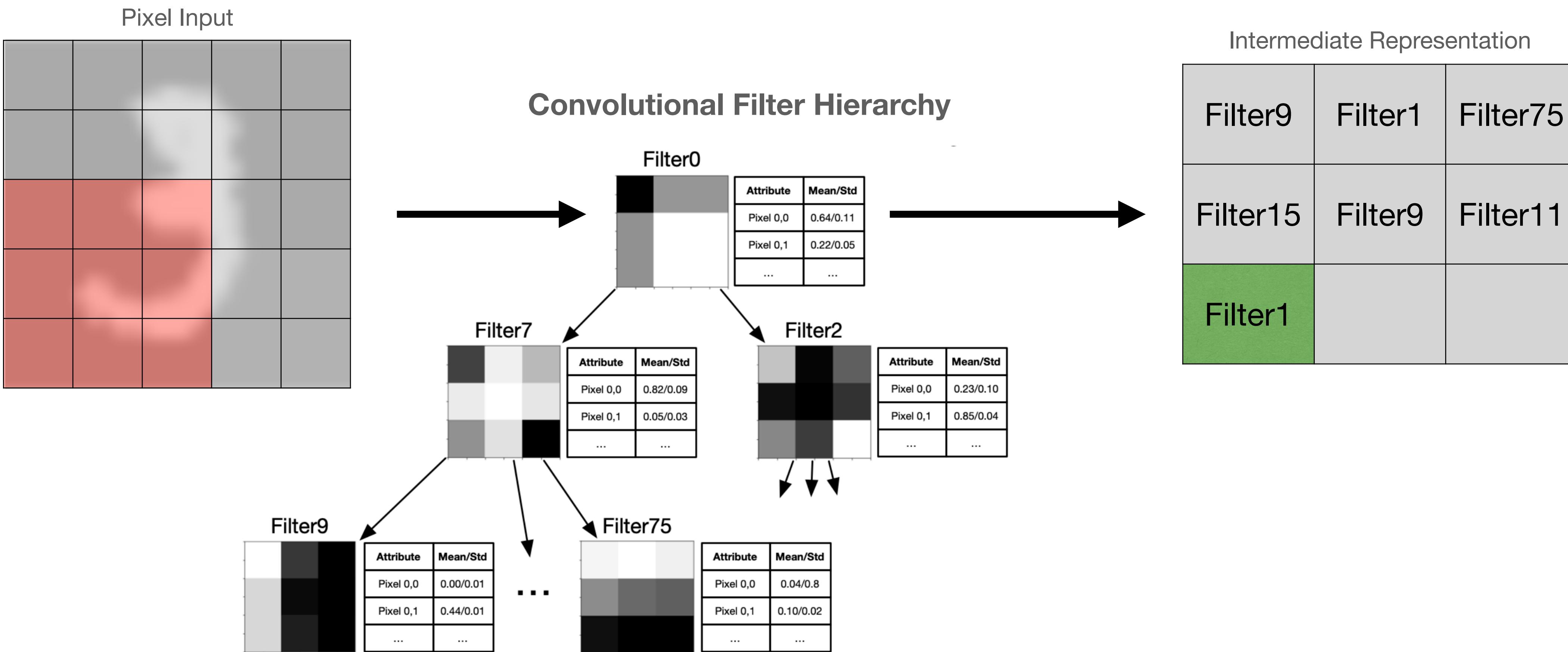
Convolutional Cobweb Approach



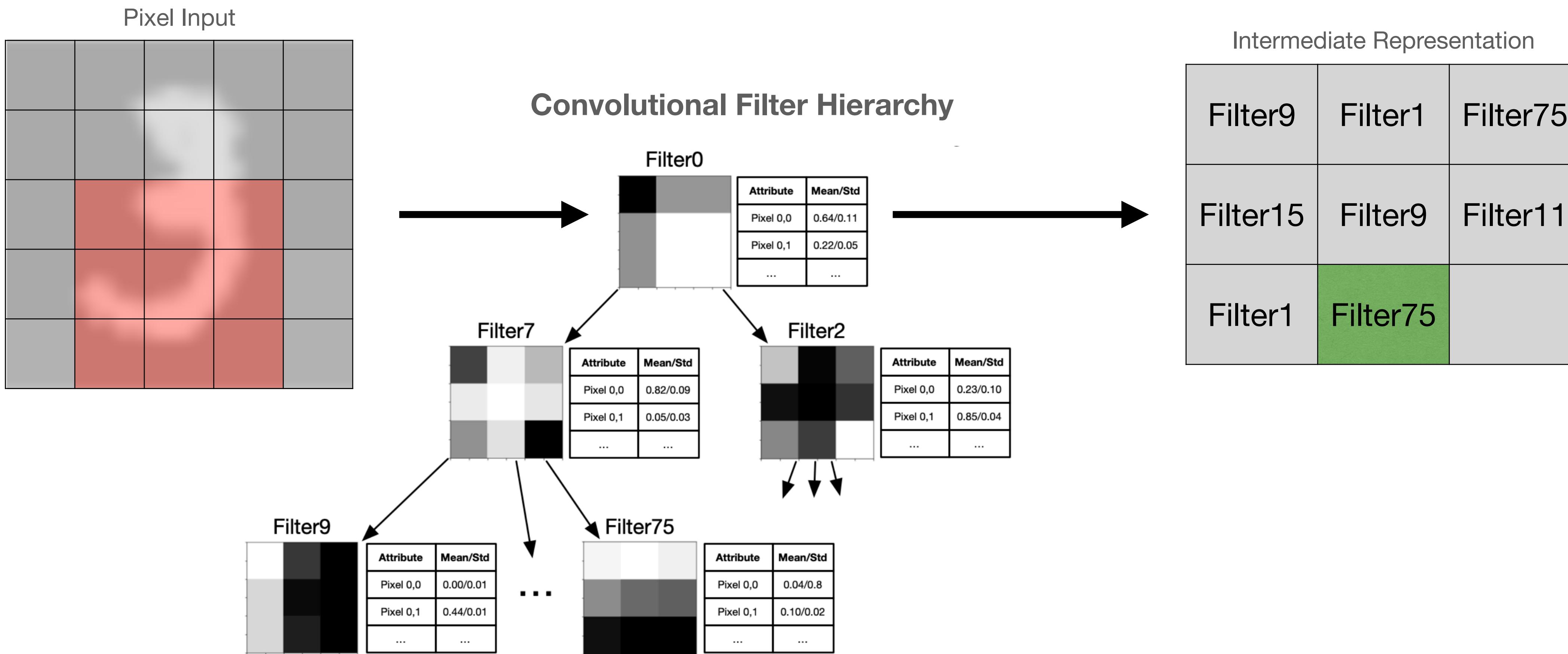
Convolutional Cobweb Approach



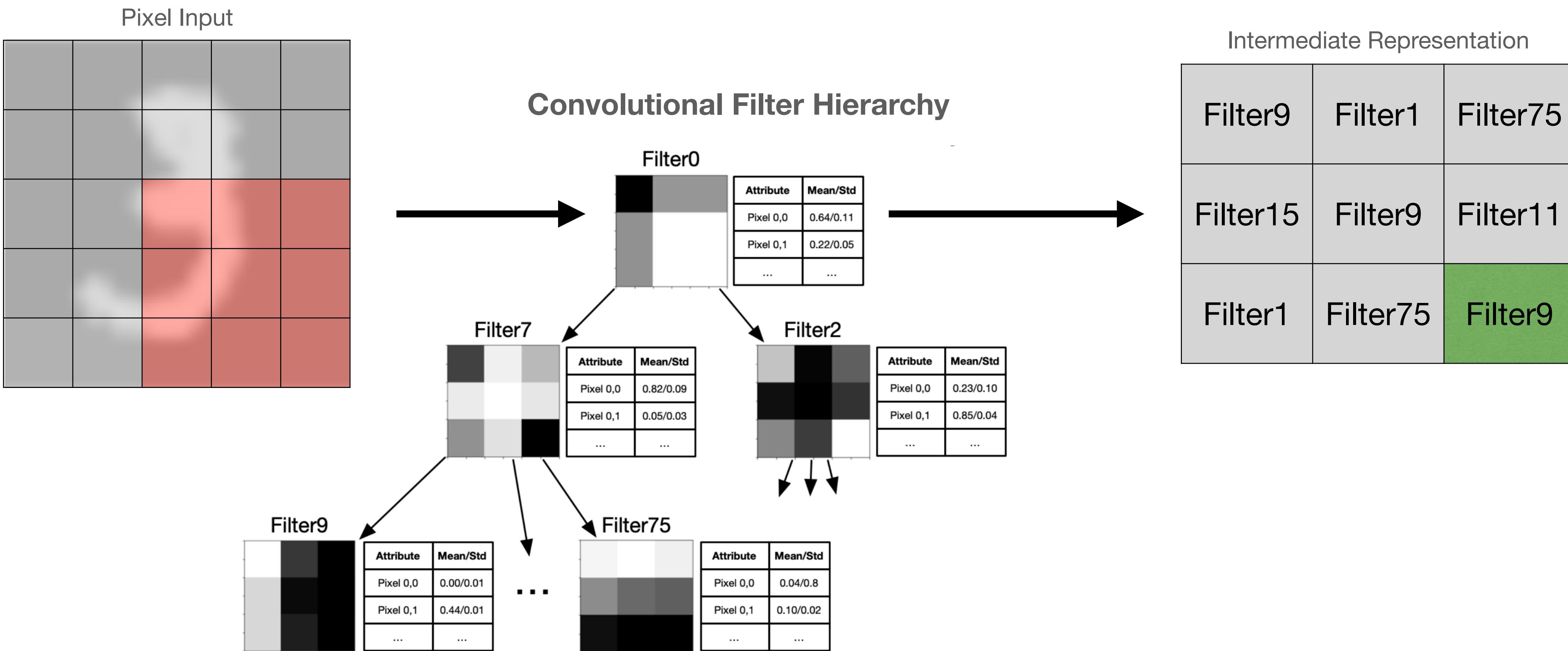
Convolutional Cobweb Approach



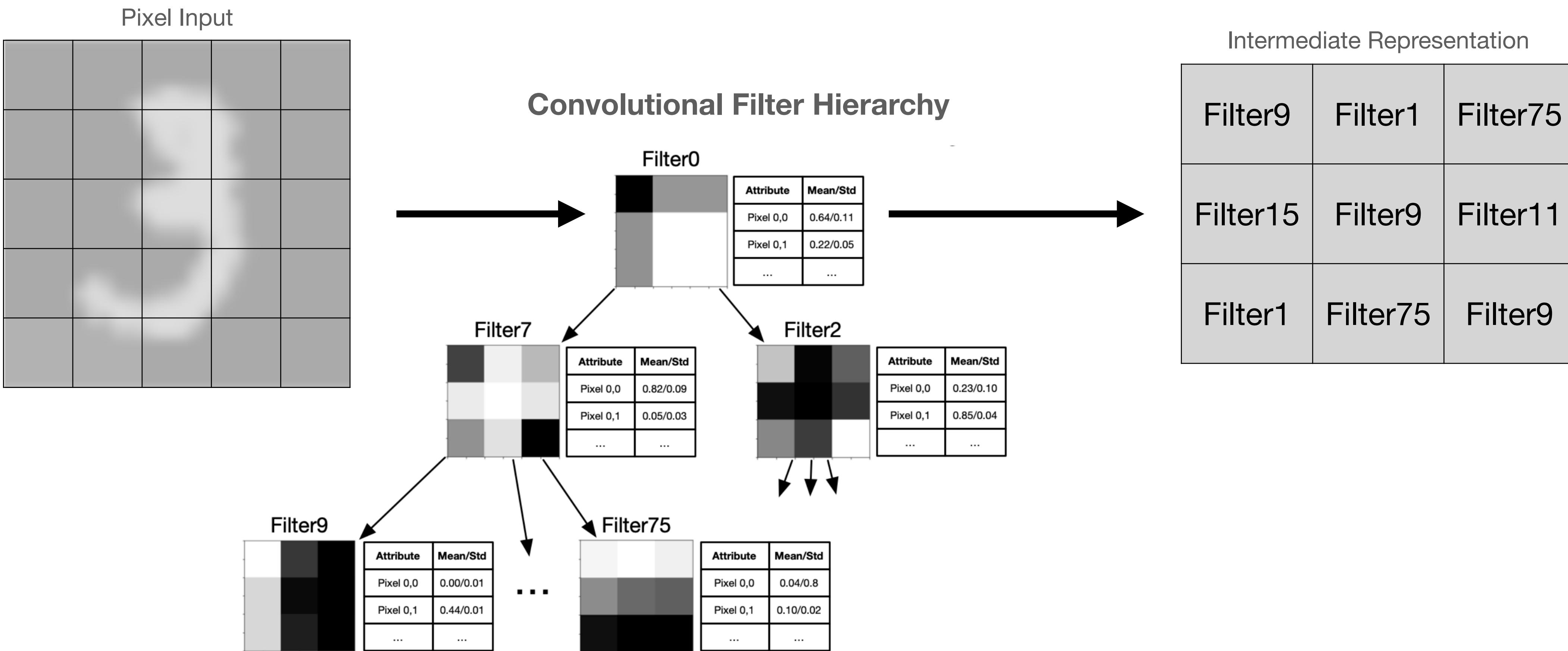
Convolutional Cobweb Approach



Convolutional Cobweb Approach



Convolutional Cobweb Approach



Convolutional Cobweb Approach

Intermediate Representation

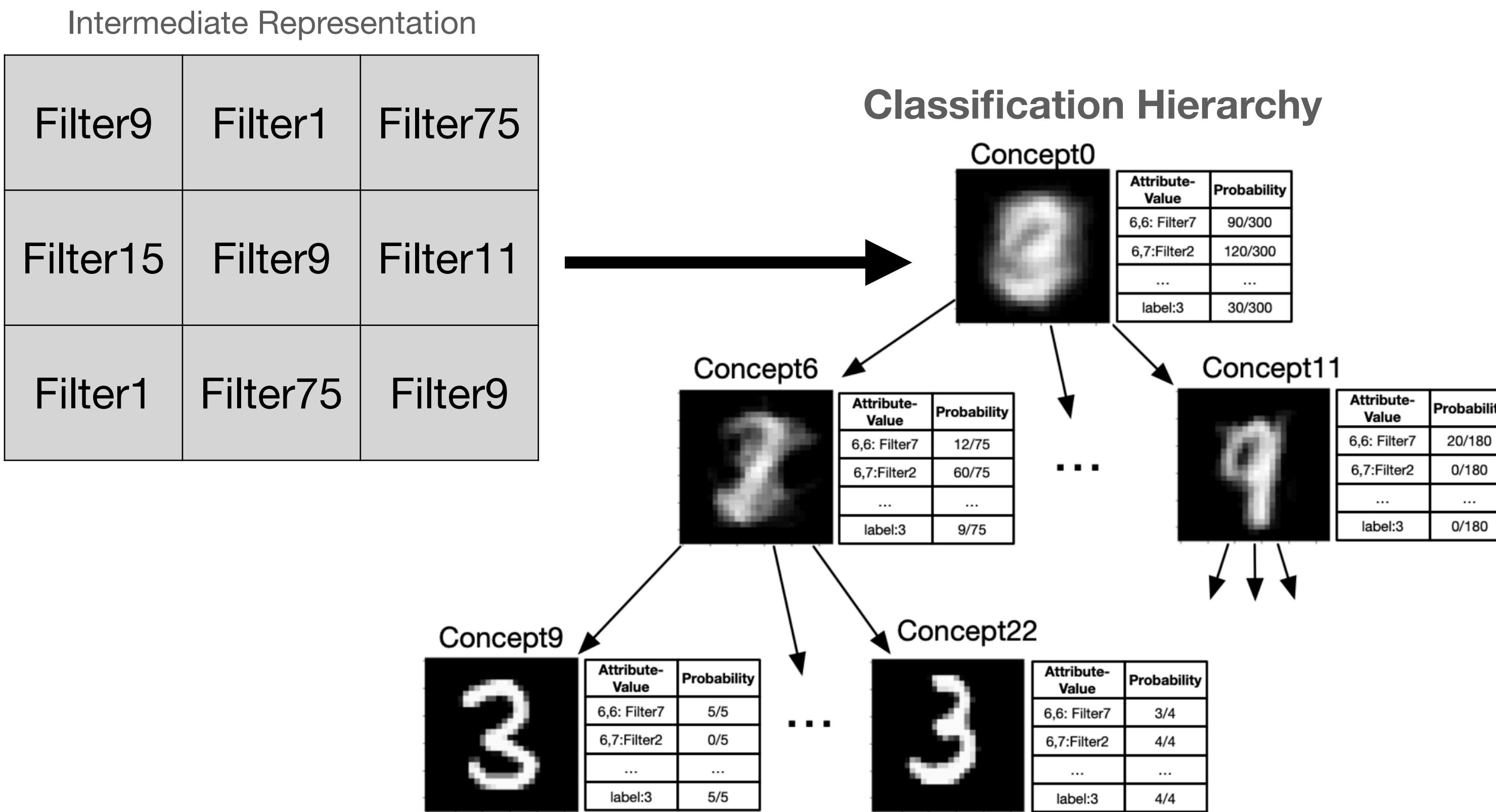
Filter9	Filter1	Filter75
Filter15	Filter9	Filter11
Filter1	Filter75	Filter9

Convolutional Cobweb Approach

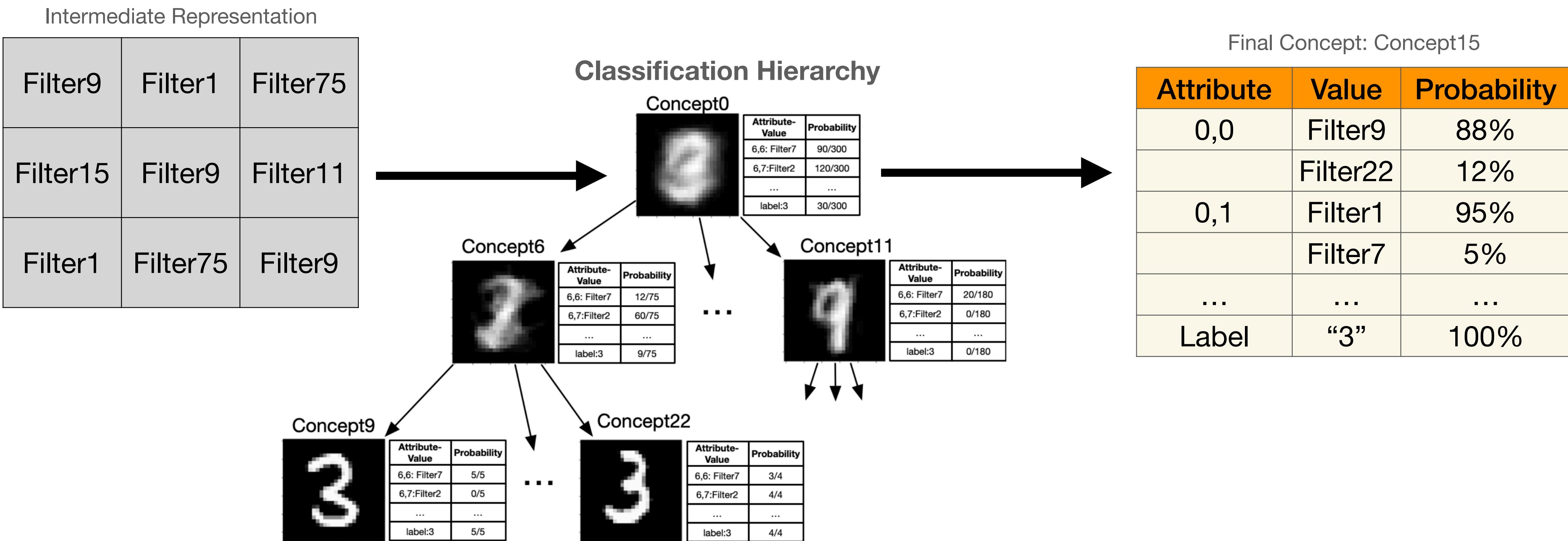
Intermediate Representation

Filter9	Filter1	Filter75
Filter15	Filter9	Filter11
Filter1	Filter75	Filter9

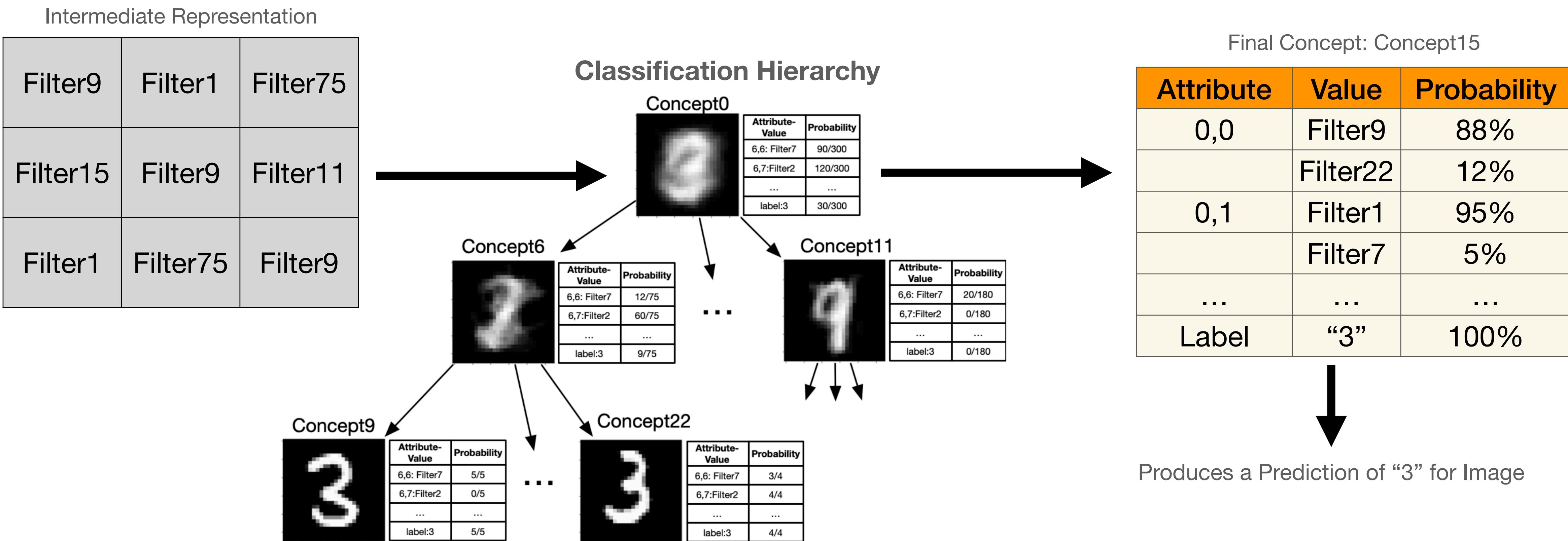
Convolutional Cobweb Approach



Convolutional Cobweb Approach



Convolutional Cobweb Approach

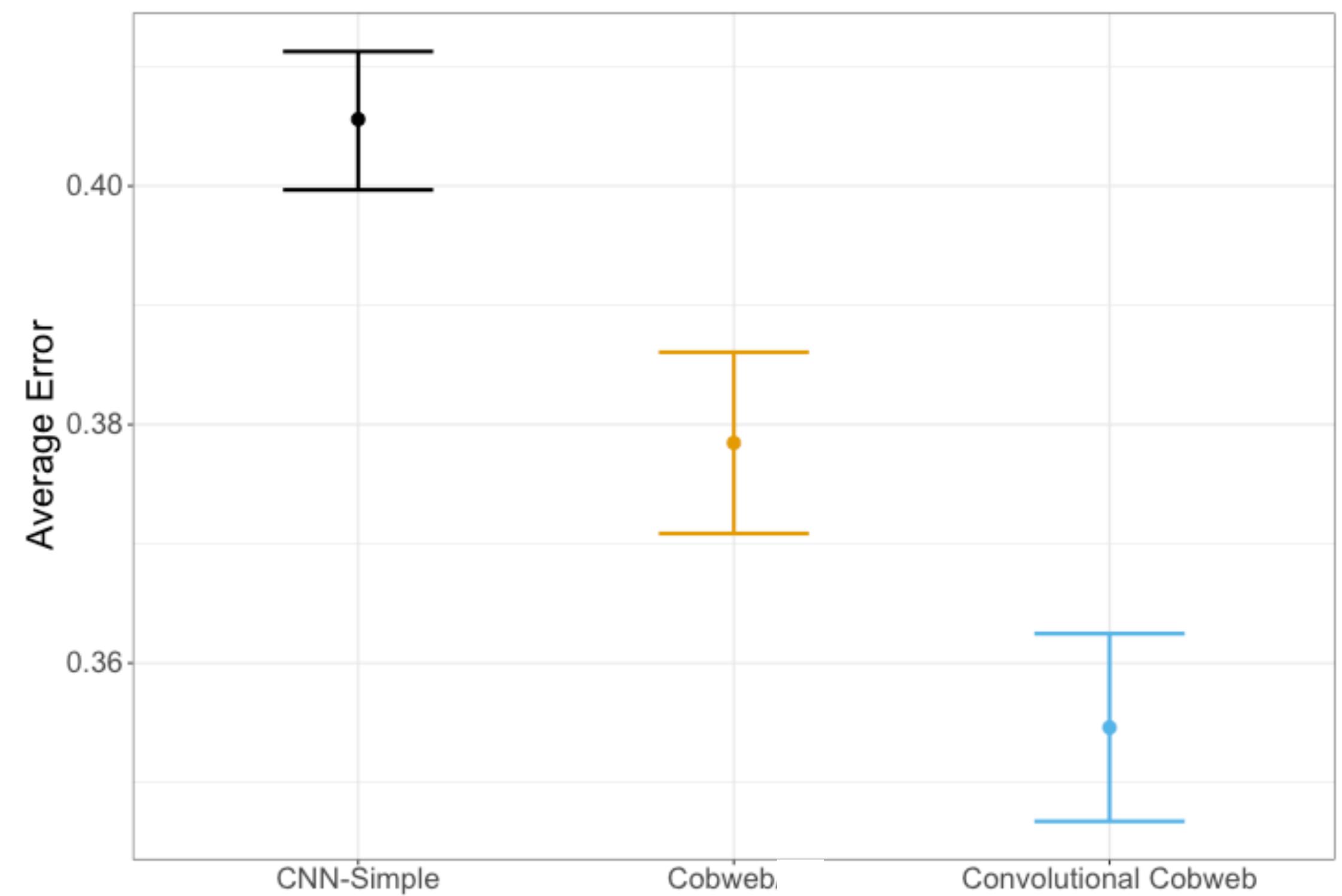


Evaluation

- As a preliminary test of our approach, we compared it to the two kinds of models we tried to unify:
 - A simple 1-layer CNN (no concept formation)
 - A Cobweb model that maps pixels to features (no convolutional filters)

Overall Performance

- Each model was applied to the incremental MNIST prediction task
- Each model was presented with 300 images (30 images for each digit)
- Images were presented in a random order (same order across models)
- Our results average over 50 runs
- We find that our approach outperforms both approaches it was based on

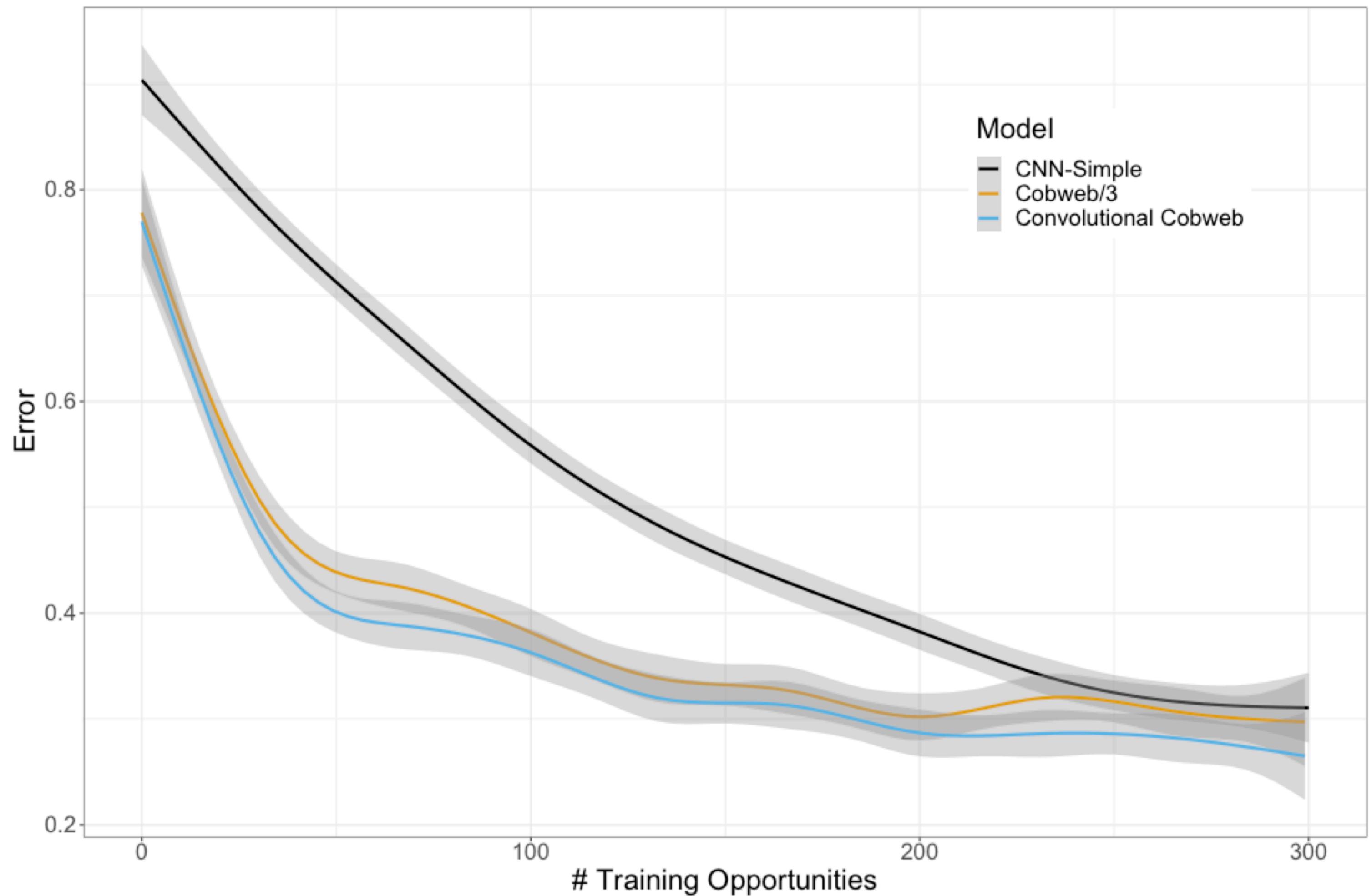


Note, CNN-Simple uses a replay buffer, given that we're training and testing incrementally.

Average performance of each model, whiskers denote bootstrapped 95% confidence intervals

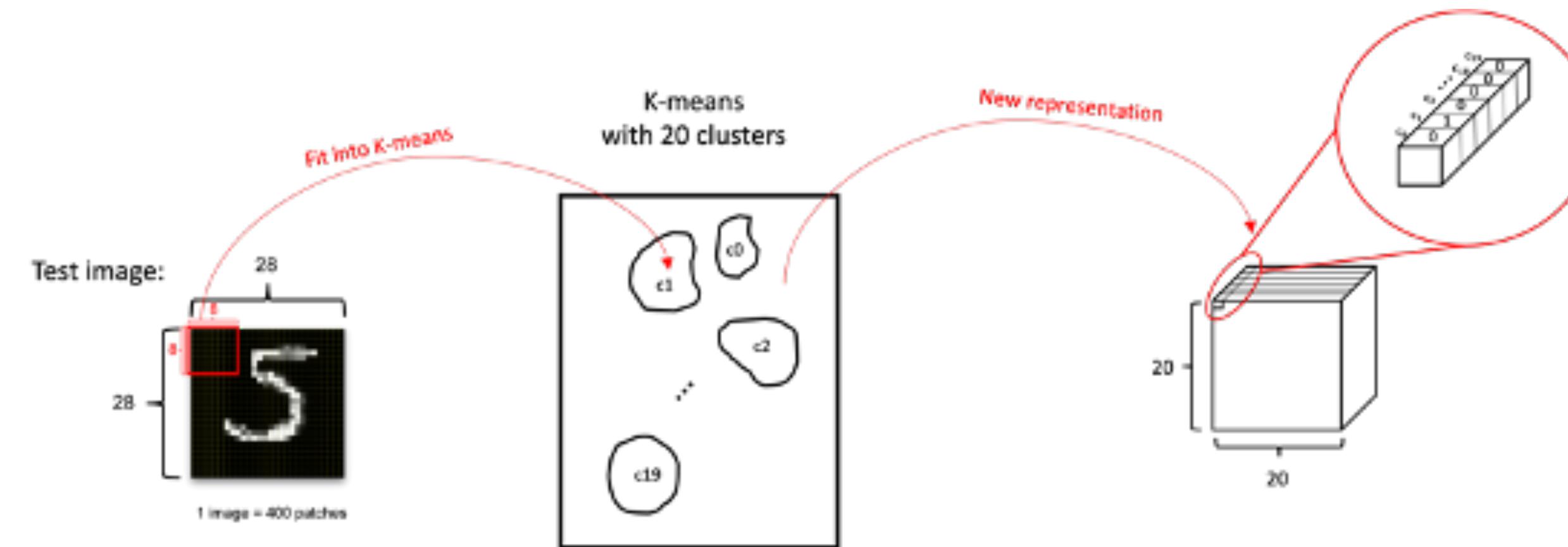
Learning Curves

- We also investigated the performance of the models over the course training
- We find that both Cobweb models seem to converge much more quickly than the CNN (likely because they're not using SGD)
- During training, our approach is only slightly better than Cobweb, but the performance is consistent over runs and across training

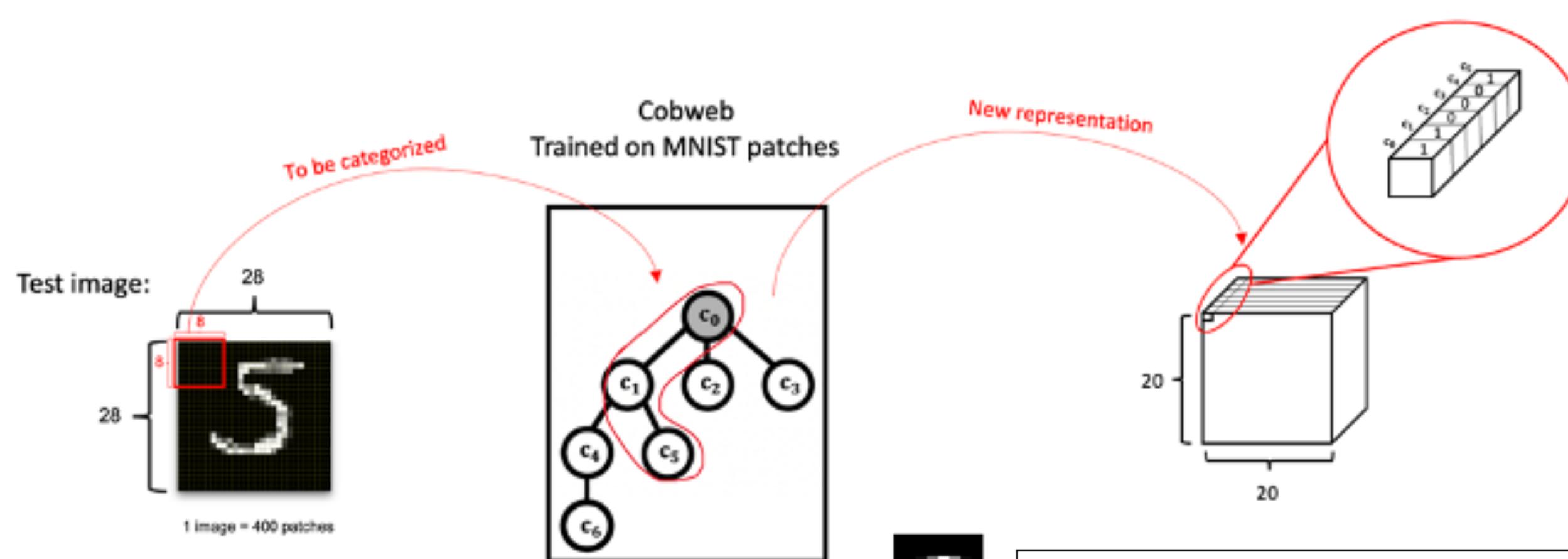


Exploring Improvements to Convolutional Processing

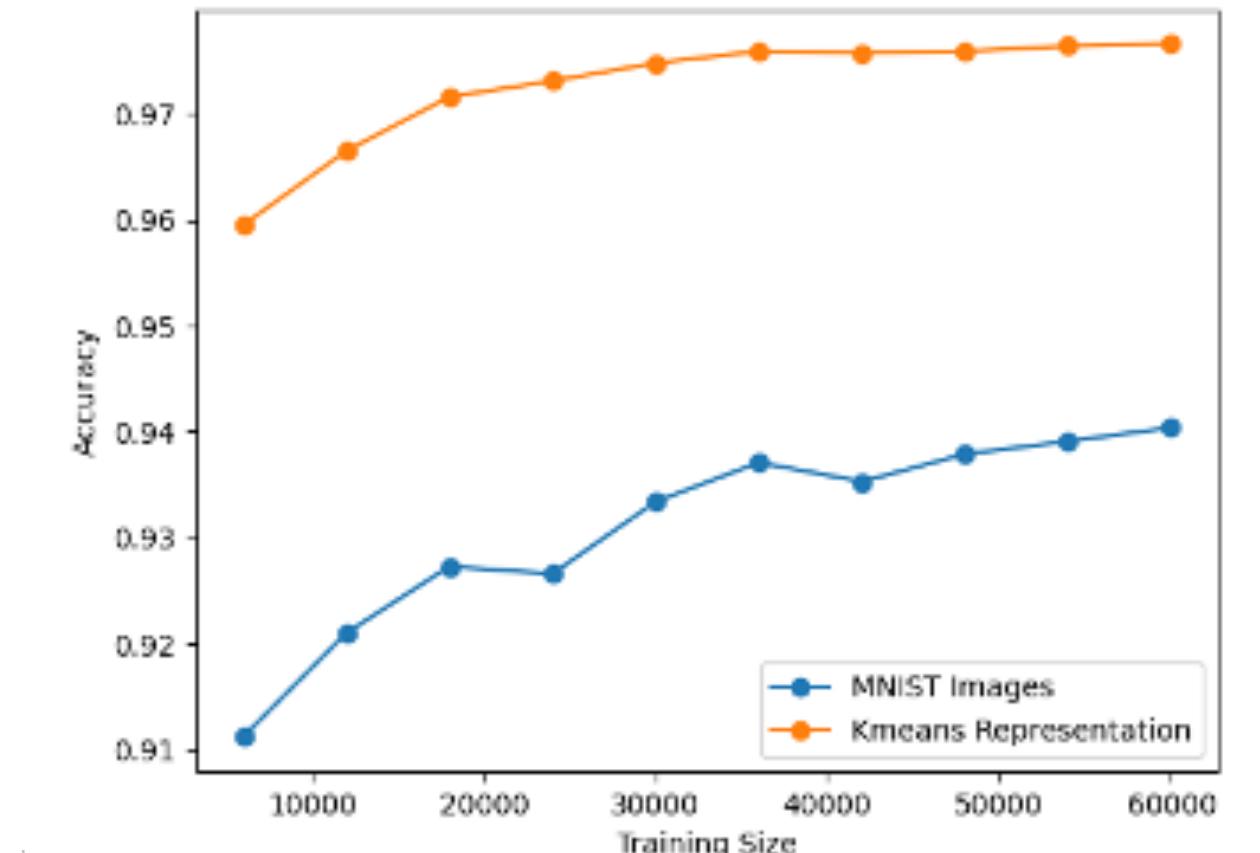
Representations generated with K-Means (single cluster label per 8x8 patch) yield increased predictive performance over using pixels directly



Currently exploring representations generated by Cobweb (multiple hierarchical labels per 8x8 patch)



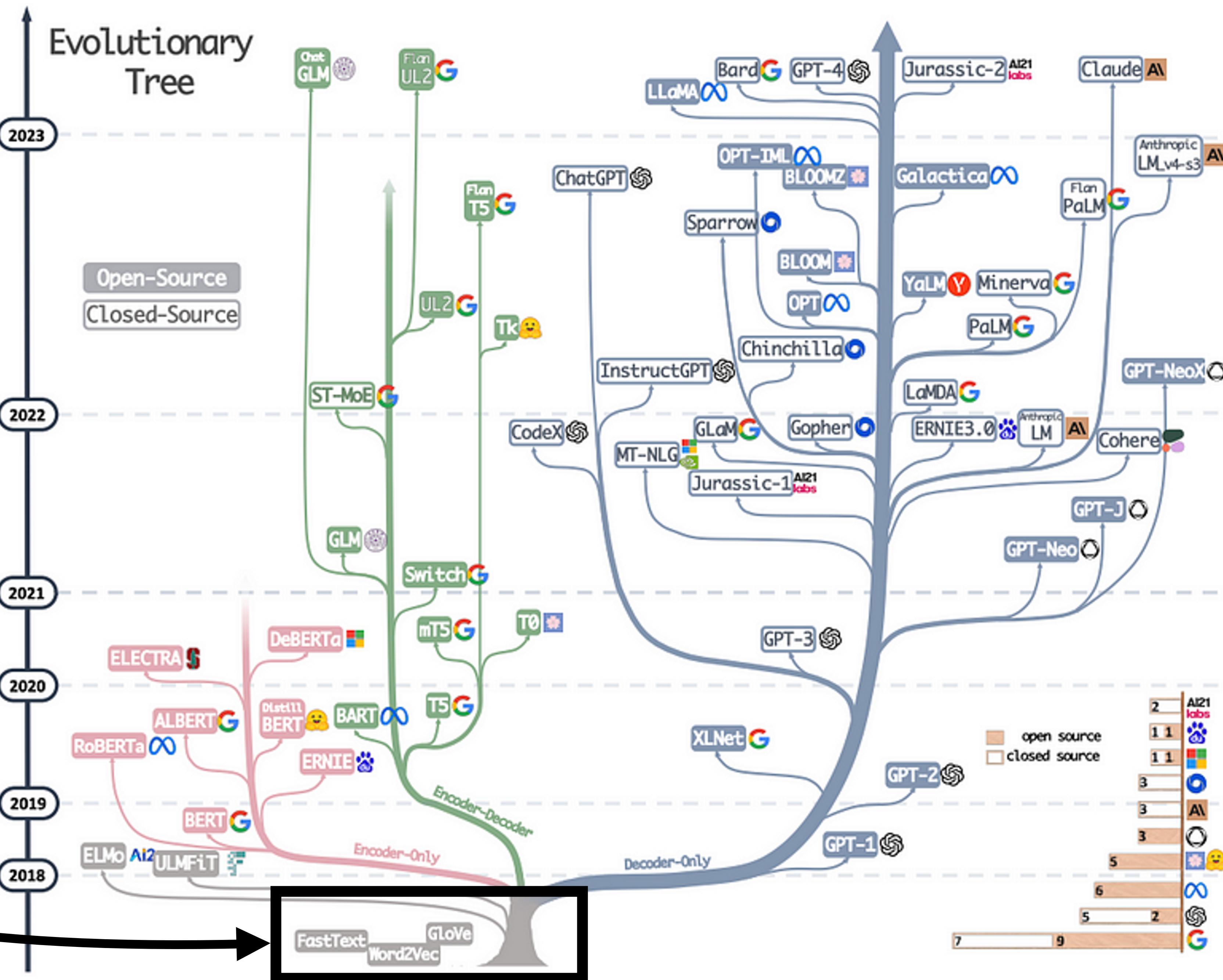
Path: 2-7014-8096-12985-13945-22297-14142-18207-86184-100810



Ongoing Research by Nicki Barari nb895@drexel.edu

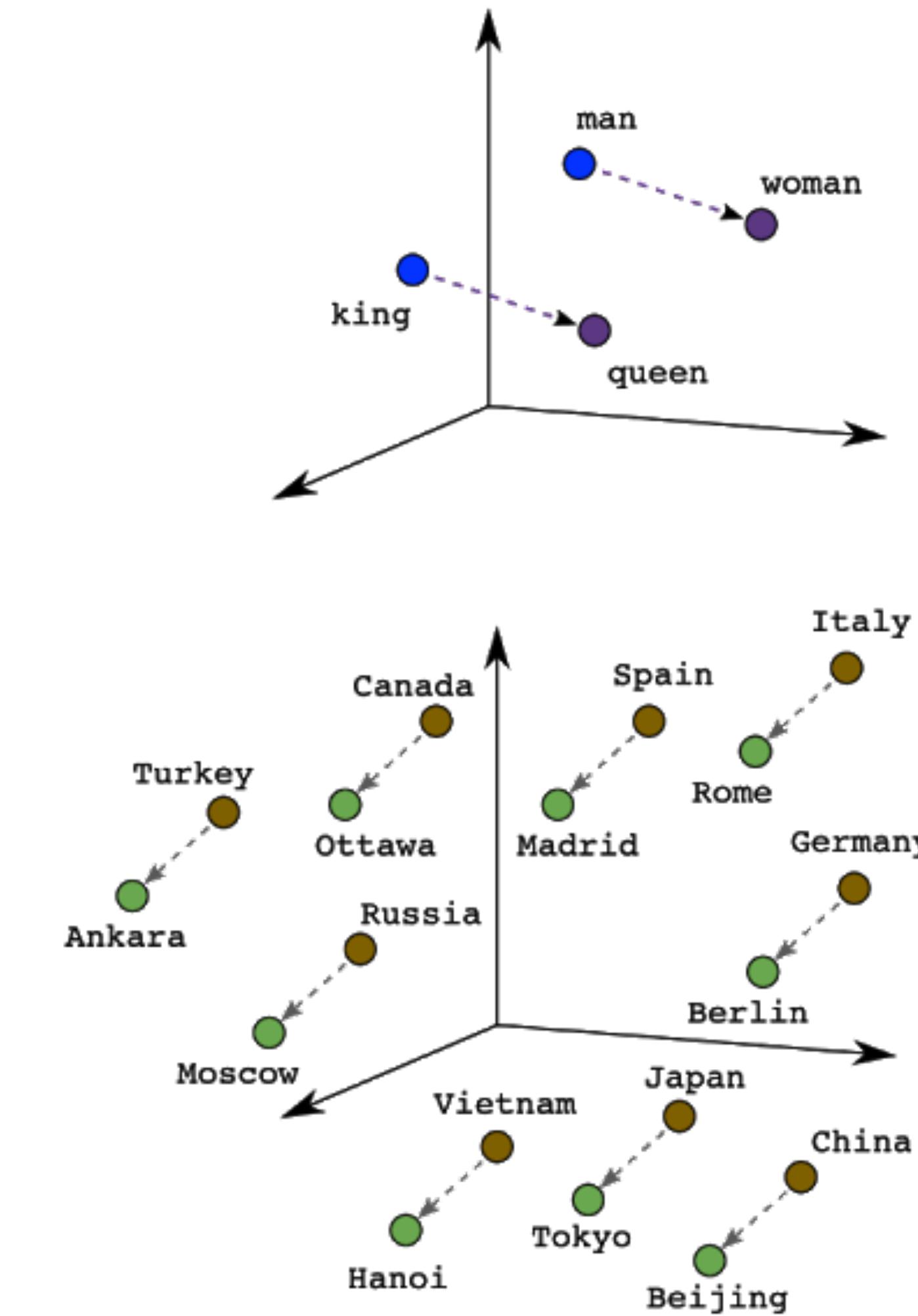
Cobweb Language Modeling Research Program

Replicate Core Capabilities of Word2Vec, Glove, etc. and
Rebase Other Approaches on Cobweb



The Basis of Modern Language Models

- The influential Word2Vec system demonstrated that one can extract meaningful semantic information in the form of word embeddings by analyzing words and their surroundings context (Mikolov et al., 2013a,b)
- This early work has grown over the past decade into the today's large language models (e.g., BERT and GPT)



Figures from: <https://daleonai.com/embeddings-explained>

Contextual Extensions to Cobweb

- Word2Vec introduced two approaches for analyzing words and their surrounding context:
 - Contextual Bag of Words (CBOW)
 - Skip-Gram
- We developed an extensions to Cobweb based on Word2Vec:
 - The Word System

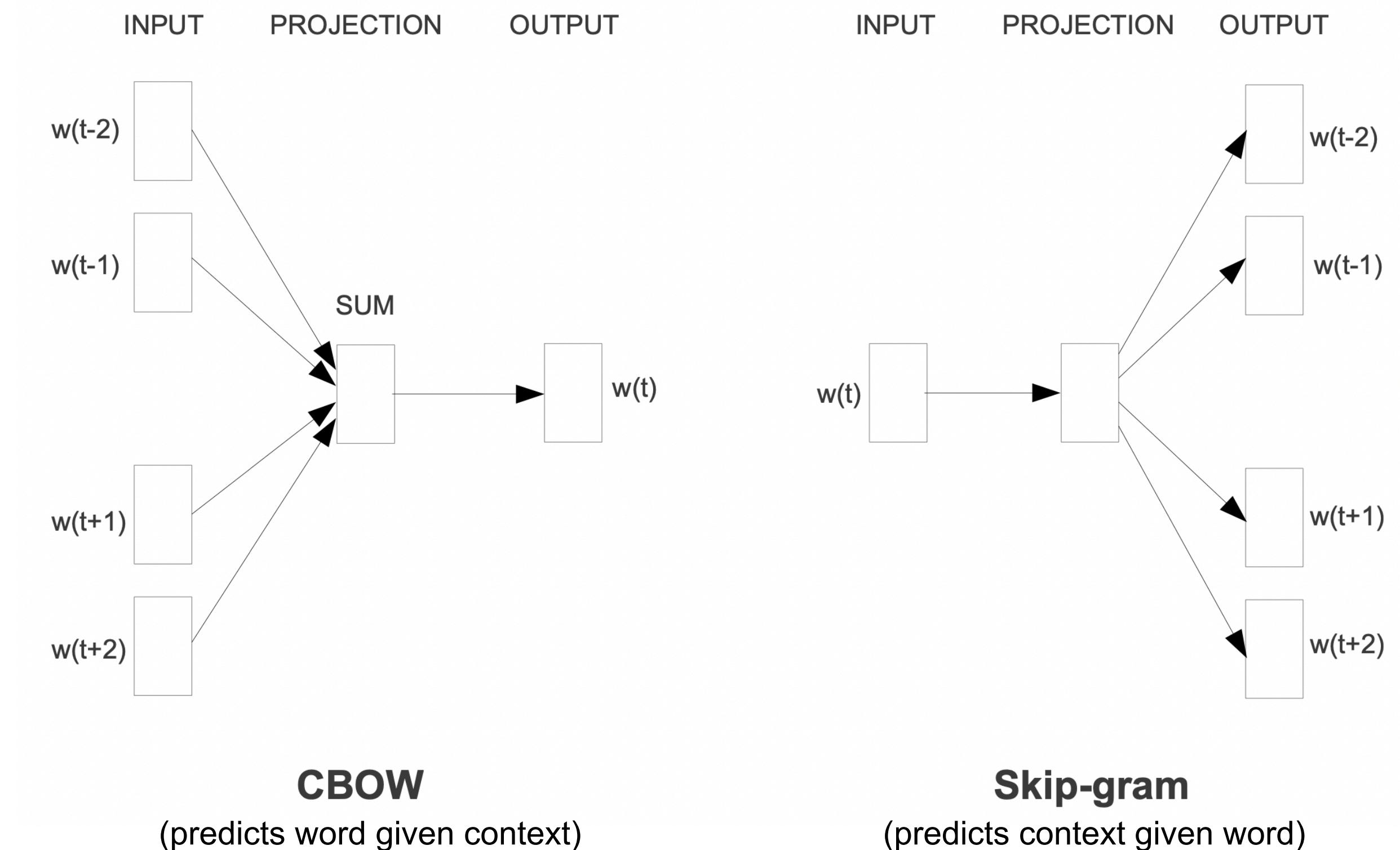


Figure from Mikolov et al. (2013): <https://arxiv.org/abs/1301.3781>

The Cobweb Word System

Example Text

I went to the house with the inspector
and saw the crime scene.

Example Text as Word Instance

Attribute	Value	$P(A=V)$
Anchor	inspector	1/1
Context	the	3/8
	house	1/8
	with	1/8
	and	1/8
	saw	1/8
	crime	1/8

The Cobweb Word System

Example Text

I went to the house with the inspector
and saw the crime scene.

Example Text as Word Instance

Attribute	Value	P(A=V)
Anchor	inspector	1/1
Context	the	3/8
	house	1/8
	with	1/8
	and	1/8
	saw	1/8
	crime	1/8

Word Concept Containing Instance

Attribute	Value	P(A=V)
Anchor	inspector	1/2
	police	1/2
Context	the	5/16
	house	1/16
	with	2/16
	and	1/16
	saw	1/16
	crime	1/16
	well	1/16
	enough	1/16
	theory	1/16
	if	1/16
	colonel	1/16

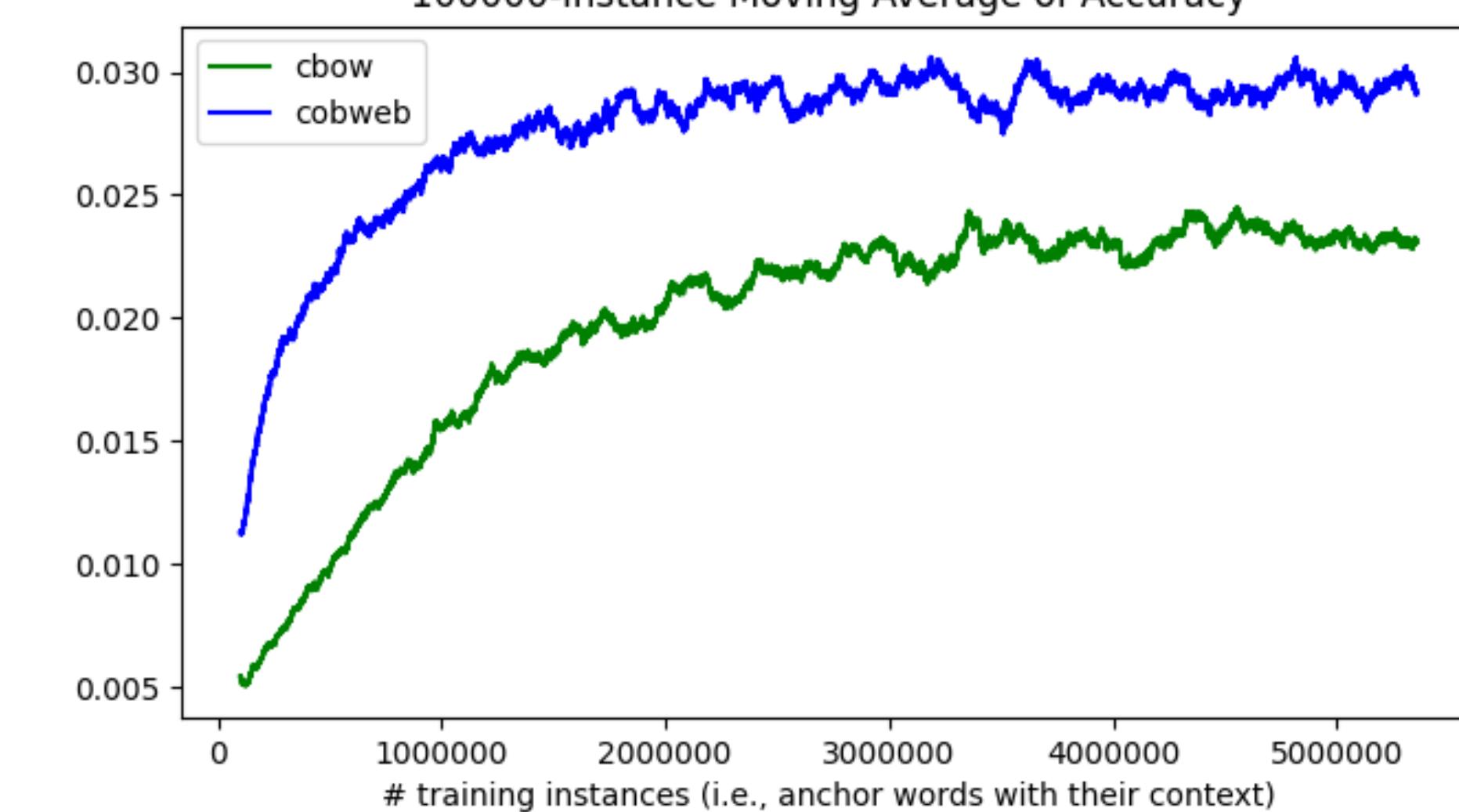
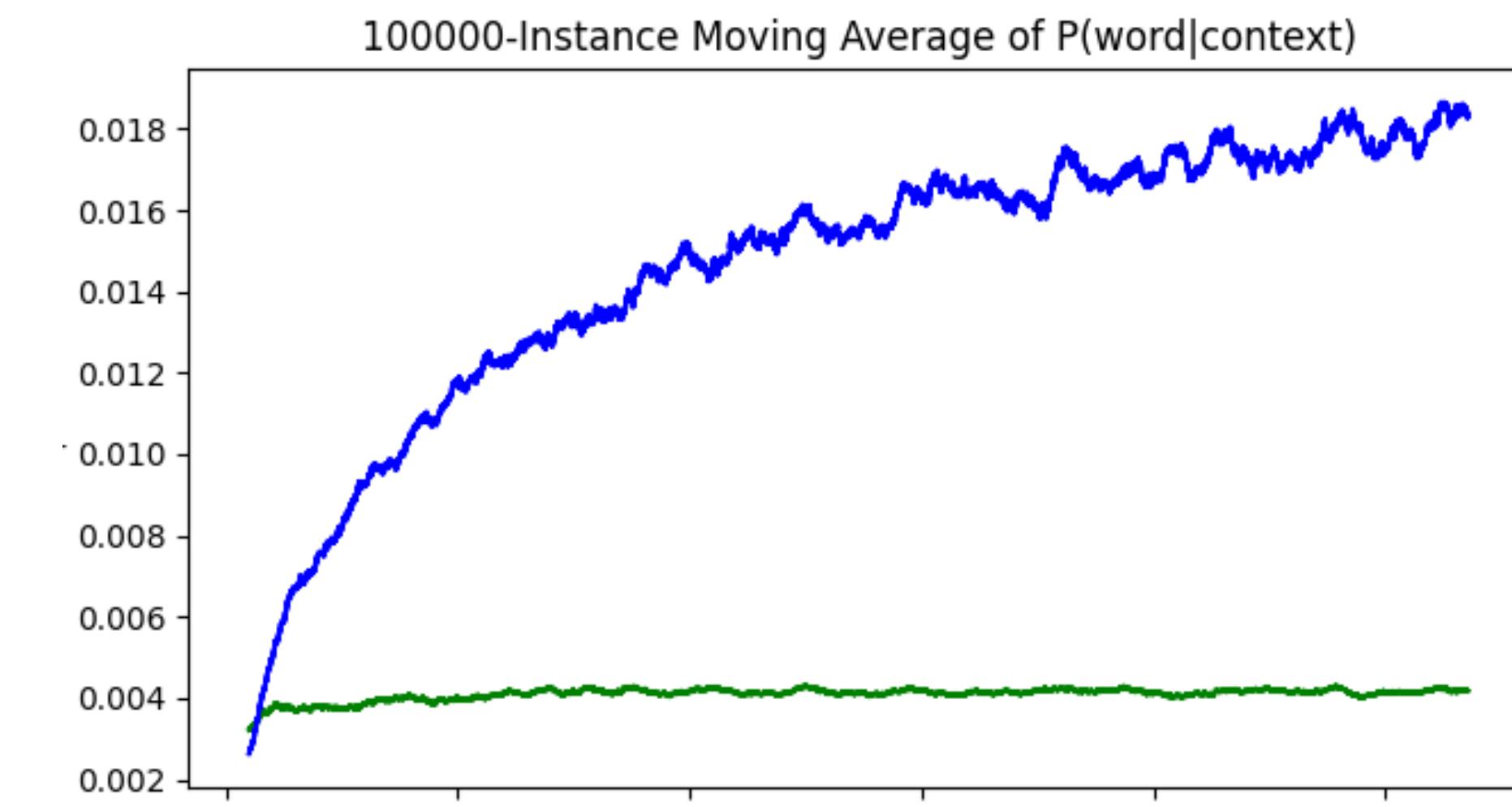
Experimental Evaluation

- We are still in the process of developing our model and scaling it up to larger data sets, but we have some preliminary results comparing our approach to Word2Vec on 500 project Gutenberg books, which are part of the Microsoft Sentence Completion Challenge data.
- To evaluate each approach, we utilized an incremental prediction paradigm where we:
 - Iterate over the corpus
 - Mask each word
 - Predict its value given the context words (10 words before and after the masked word)
 - Evaluate the prediction
 - Update the model by training it with the true anchor word and its context

Experimental Evaluation

Cobweb Outperforms Word2Vec's CBOW Approach

- Our Cobweb model expands 100 nodes to make a prediction
- We use the CBOW variant of Word2Vec (the variant that *can* predict anchor given context)
- Our results suggest that Cobweb improves at predicting the anchor word much more quickly than Word2Vec
- Additionally, it achieves better predictive performance overall



An Analysis of Training Cost

Cobweb Scales Better than Word2Vec

MacLellan, C.J., Matsakis, P., & Langley, P. (2022). Efficient Induction of Language Models via Probabilistic Concept Formation. In Proceedings of the Tenth Annual Conference on Advances in Cognitive Systems.

- Cobweb can efficiently update both its structure and its parameters
 - Its asymptotic complexity is better than Word2Vec, roughly $O(n \log n)$ vs $O(n^2)$
 - Cobweb can efficiently update its parameters without retraining on prior data
 - As demonstrated earlier, it is robust to catastrophic forgetting

Approach	Asymptotic Run Time
Word2Vec-CBOW	$O(N^2 \times E \times (W \times D + D \times \log_2(V)))$
Cobweb-Word	$O(N \times B^2 \times \log_B(N) \times V)$

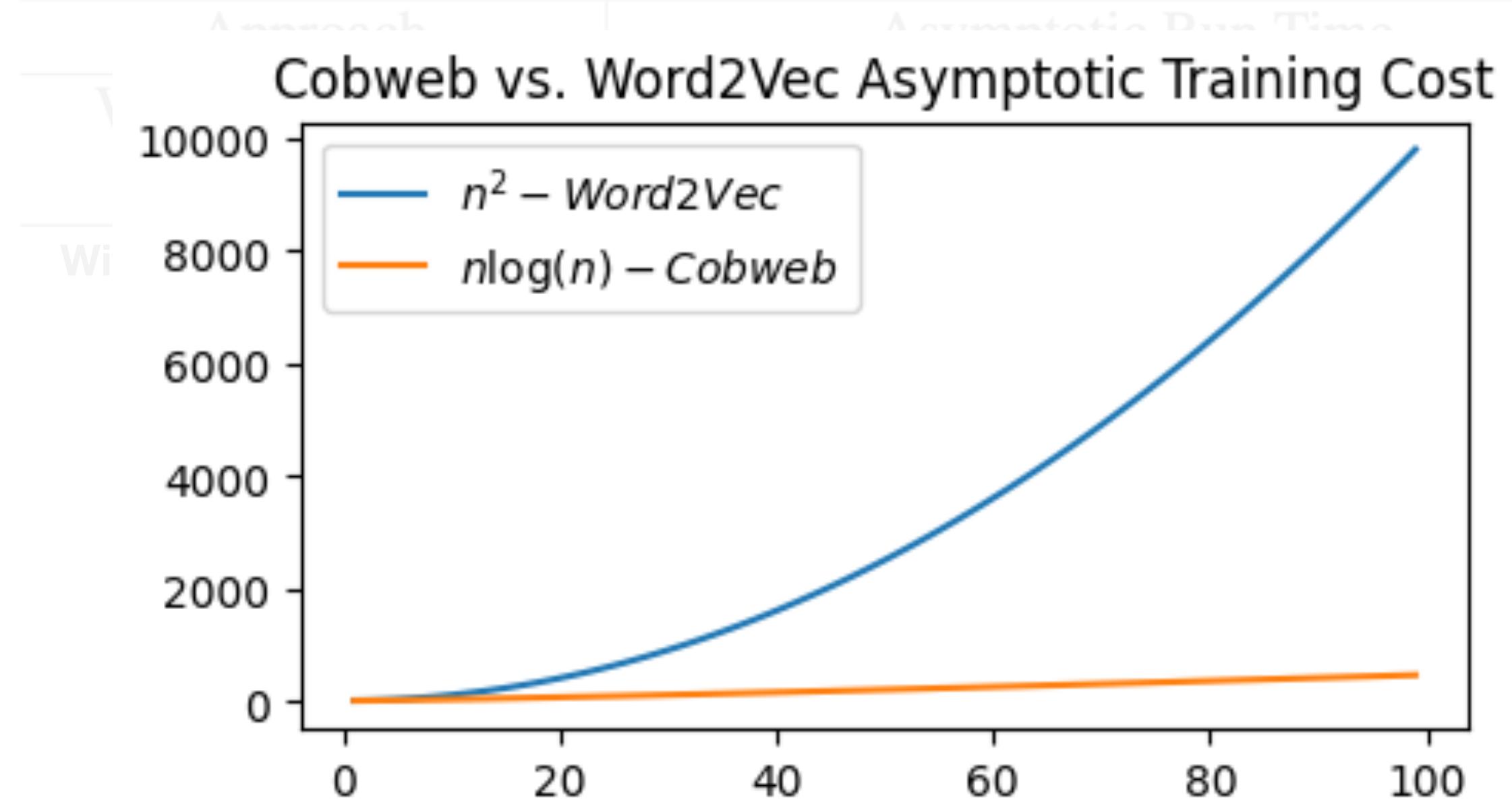
Cobweb with recent improvements:
 N is number of words in corpus
 E is number of epochs
 W is the size of the window
 D is the dimensionality of the embedding
 V is the size of the vocabulary
 B is the branching factor of the cobweb tree
 C is the number of concepts in the tree

An Analysis of Training Cost

Cobweb Scales Better than Word2Vec

MacLellan, C.J., Matsakis, P., & Langley, P. (2022). Efficient Induction of Language Models via Probabilistic Concept Formation. In Proceedings of the Tenth Annual Conference on Advances in Cognitive Systems.

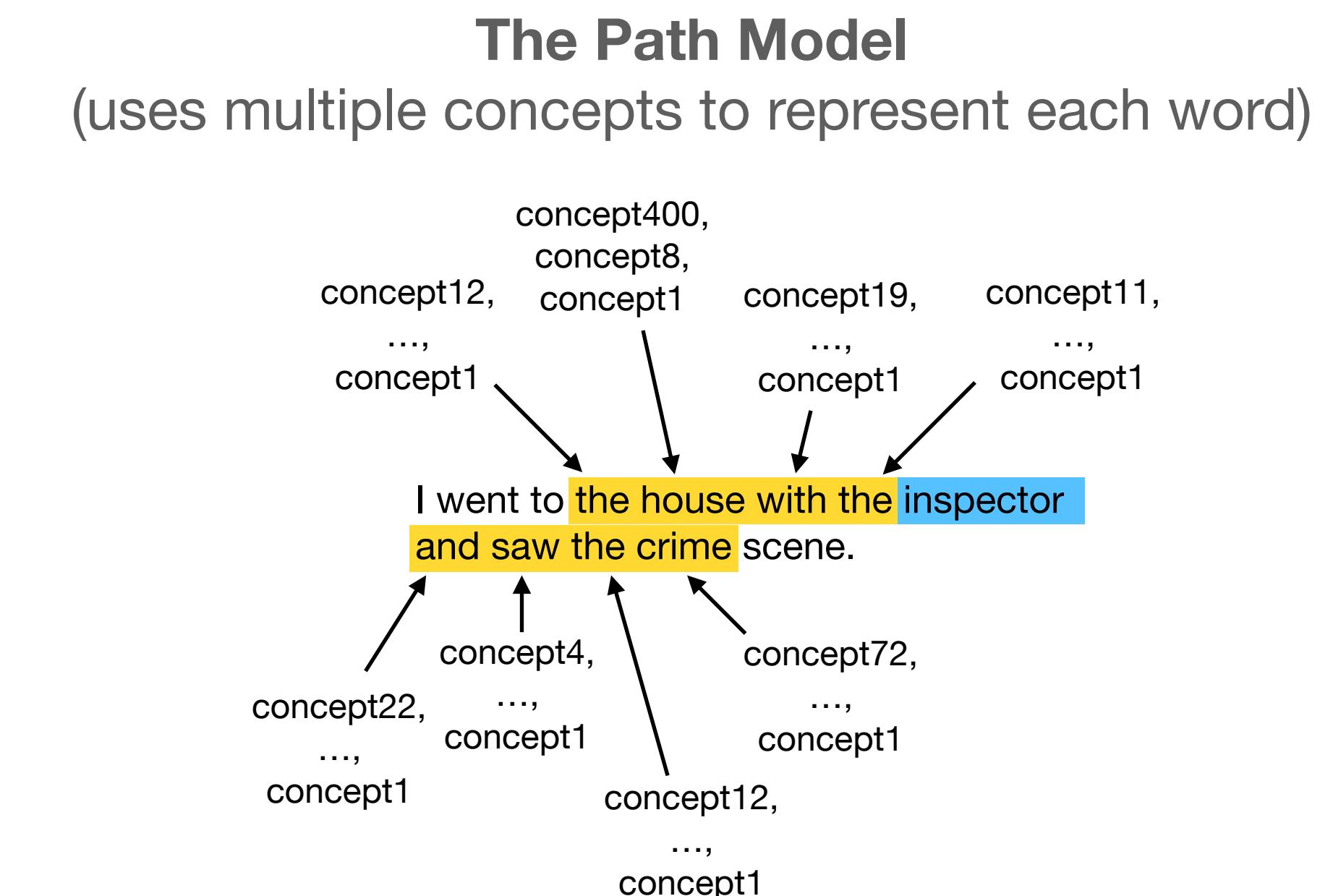
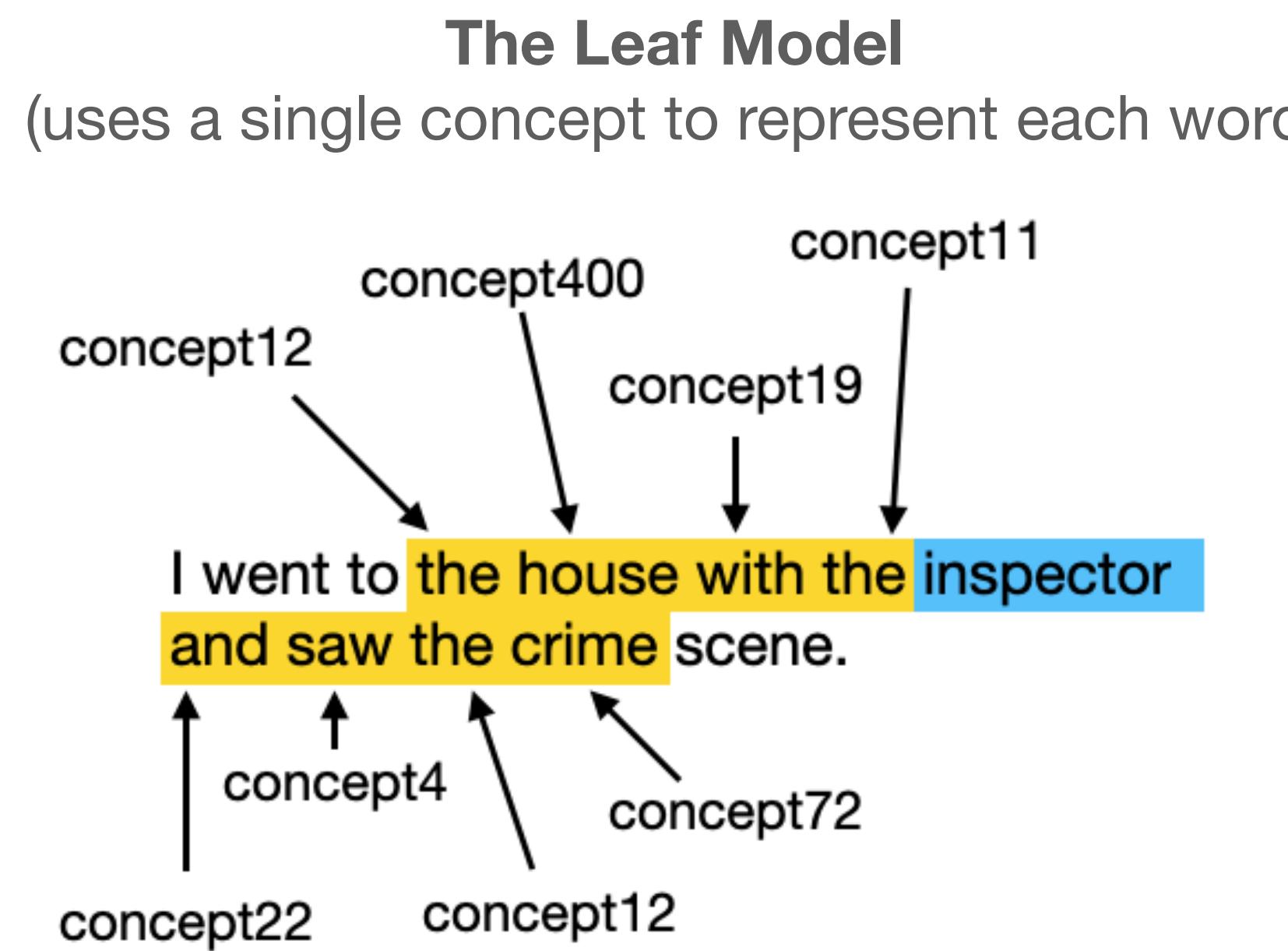
- Cobweb can efficiently update both its structure and its parameters
 - Its asymptotic complexity is better than Word2Vec, roughly $O(n \log n)$ vs $O(n^2)$
 - Cobweb can efficiently update its parameters without retraining on prior data
 - As demonstrated earlier, it is robust to catastrophic forgetting



B is the branching factor of the cobweb tree
Practically, we just finished training the largest Cobweb model ever built with over over 14 million instances (~500 books)!

Next Steps

Learning Intermediate Representations



MacLellan, C.J., Matsakis, P., & Langley, P. (2022). Efficient Induction of Language Models via Probabilistic Concept Formation. In Proceedings of the Tenth Annual Conference on Advances in Cognitive Systems. [\(pdf\)](#) [\(talk\)](#)

General Discussion

- We have been developing foundational Cobweb building blocks, so that we can demonstrate efficient, scalable, and high-performance capabilities.
- Cobweb is well-suited for incremental learning across a wide range of tasks and domains.
- We believe it has the potential to be competitive with deep learning, while retaining many of its benefits (e.g., data efficiency and robustness to forgetting)

Thank you!

New and Future Extensions to Cobweb

Christopher J. MacLellan

<https://chrismaclellan.com>

cmaclell@gatech.edu



<https://tail.cc.gatech.edu>