Jye Sawtell-Rickson
m1361019@cgu.edu.tw

Chang Gung University

April 30, 2026

Paper Review: DreamCoder

Growing generalizable, interpretable knowledge with wake-sleep Bayesian program learning, MIT

Outline

m13610190@cgu.edu.tw

Input/Output Problems

Basic problem: what is the function that maps input to output?

- $[7, 2, 3] \rightarrow [4, 3, 8]$
- $[3, 8] \longrightarrow [9, 4]$
- $[4, 3, 2] \rightarrow [3, 4, 5]$
- $[1, 3, 0] \rightarrow ?$

Solution:

- Add one and reverse
- each(plus1) $\rightarrow$ reverse

Key Question

How can we automatically discover this function?

m1361019@cgu.edu.tw

Input/Output Problems

Basic problem: what is the function that maps input to output?

- $[7, 2, 3] \rightarrow [4, 3, 8]$
- $[3, 8] \longrightarrow [9, 4]$
- $[4, 3, 2] \rightarrow [3, 4, 5]$
- $[1, 3, 0] \rightarrow ?$

Solution:

- Add one and reverse
- each(plus1) $\rightarrow$ reverse

Key Question

How can we automatically discover this function?

m1361019@cgu.edu.tw

Input/Output Problems

Basic problem: what is the function that maps input to output?

- $[7, 2, 3] \rightarrow [4, 3, 8]$
- $[3, 8] \longrightarrow [9, 4]$
- $[4, 3, 2] \rightarrow [3, 4, 5]$
- $[1, 3, 0] \rightarrow ?$

Solution:

- Add one and reverse
- each(plus1) $\rightarrow$ reverse

Key Question

How can we automatically discover this function?

m1361019@cgu.edu.tw

Input/Output Problems

Basic problem: what is the function that maps input to output?

- $[7, 2, 3] \rightarrow [4, 3, 8]$
- $[3, 8] \longrightarrow [9, 4]$
- $[4, 3, 2] \rightarrow [3, 4, 5]$
- $[1, 3, 0] \rightarrow ?$

Solution:

- Add one and reverse
- each(plus1) $\rightarrow$ reverse

Key Question

How can we automatically discover this function?

m13610190@cgu.edu.tw

Introduction

- Program synthesis: search for programs.
- Applications: FlashFill, AlphaCode.
- We need to search efficiently (Chollet's measure of intelligence)
- Dreamcoder proposes new algorithm.

m1361019@cgu.edu.tw

Introduction

- Program synthesis: search for programs.
- Applications: FlashFill, AlphaCode.
- We need to search efficiently (Chollet's measure of intelligence)
- Dreamcoder proposes new algorithm.

m1361019@cgu.edu.tw

Introduction

- Program synthesis: search for programs.
- Applications: FlashFill, AlphaCode.
- We need to search efficiently (Chollet's measure of intelligence)
- Dreamcoder proposes new algorithm.

m13610190@cgu.edu.tw

Introduction

- Program synthesis: search for programs.
- Applications: FlashFill, AlphaCode.
- We need to search efficiently (Chollet's measure of intelligence)
- Dreamcoder proposes new algorithm.

m1361019@cgu.edu.tw

Key Contributions

- Development of a system that solves problems by writing programs.
- A Bayesian 'wake-sleep' learning algorithm for efficient learning.
- Discover interpretable, reusable, and generalisable knowledge.
- Demonstration of success across multiple domains.

m13610190@cgu.edu.tw

Key Contributions

- Development of a system that solves problems by writing programs.
- A Bayesian 'wake-sleep' learning algorithm for efficient learning.
- Discover interpretable, reusable, and generalisable knowledge.
- Demonstration of success across multiple domains.

m1361019@cgu.edu.tw

Key Contributions

- Development of a system that solves problems by writing programs.
- A Bayesian 'wake-sleep' learning algorithm for efficient learning.
- Discover interpretable, reusable, and generalisable knowledge.
- Demonstration of success across multiple domains.

m13610190@cgu.edu.tw

Key Contributions

- Development of a system that solves problems by writing programs.
- A Bayesian 'wake-sleep' learning algorithm for efficient learning.
- Discover interpretable, reusable, and generalisable knowledge.
- Demonstration of success across multiple domains.

m13610190@cgu.edu.tw

Problem Statement

Using a library of basis functions, **search** for a function that solves some specification.

Key difficulties

1. To be effective, basis functions are hand-crafted, giving a strong inductive bias.
2. Search space grows exponentially and is expensive.

DreamCoder addresses both of these with their wake-sleep algorithm.

m13610190@cgu.edu.tw

Problem Statement

Using a library of basis functions, **search** for a function that solves some specification.

Key difficulties

1. To be effective, basis functions are hand-crafted, giving a strong inductive bias.
2. Search space grows exponentially and is expensive.

DreamCoder addresses both of these with their wake-sleep algorithm.

m1361019@cgu.edu.tw

Problem Statement

Using a library of basis functions, **search** for a function that solves some specification.

Key difficulties

1. To be effective, basis functions are hand-crafted, giving a strong inductive bias.
2. Search space grows exponentially and is expensive.

DreamCoder addresses both of these with their wake-sleep algorithm.

m1361019@cgu.edu.tw

Problem Statement

Using a library of basis functions, **search** for a function that solves some specification.

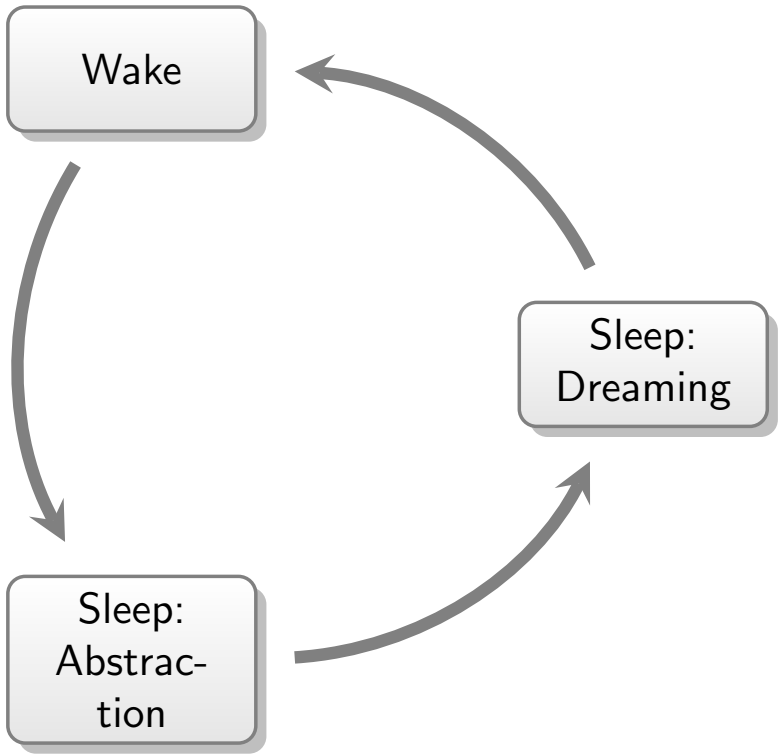
Key difficulties

1. To be effective, basis functions are hand-crafted, giving a strong inductive bias.
2. Search space grows exponentially and is expensive.

DreamCoder addresses both of these with their wake-sleep algorithm.

m1361019@cgu.edu.tw

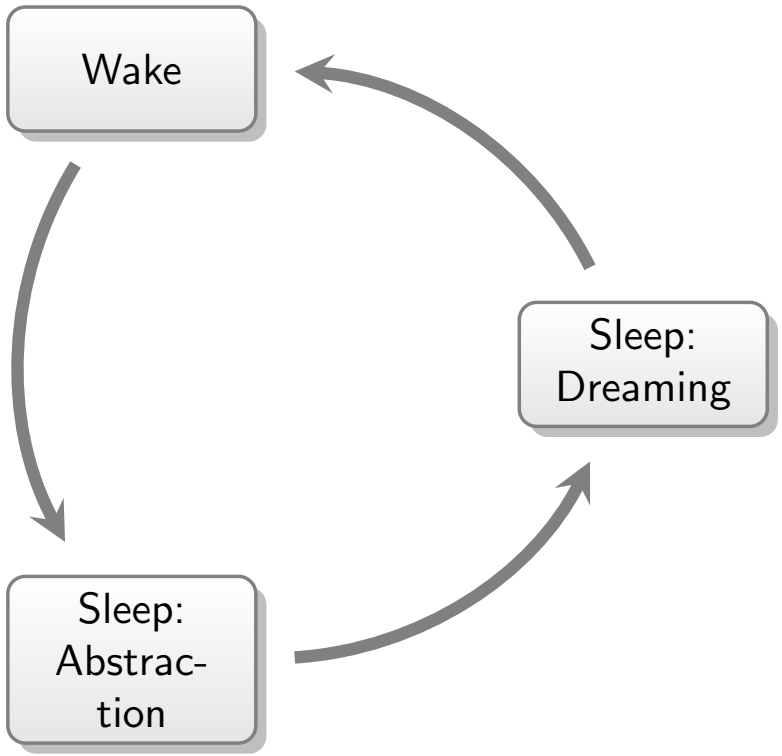
Wake-Sleep Learning



1. **Wake**: find the best program given the current library.
2. **Abstraction**: analyse and compress programs, grow library.
3. **Dreaming**: train a recognition model to predict programs.

m13610190@cgu.edu.tw

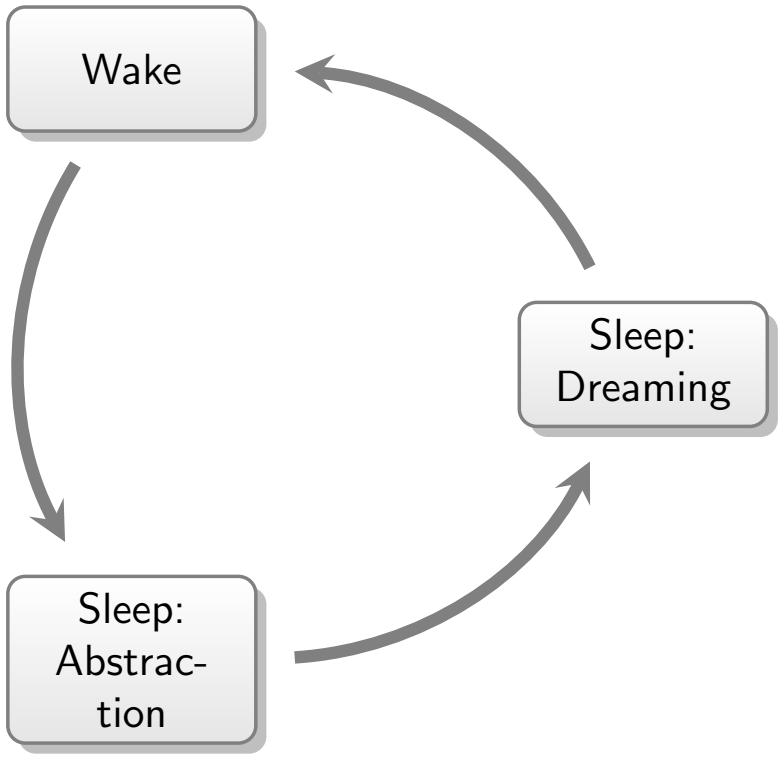
Wake-Sleep Learning



1. **Wake**: find the best program given the current library.
2. **Abstraction**: analyse and compress programs, grow library.
3. **Dreaming**: train a recognition model to predict programs.

m13610190@cgu.edu.tw

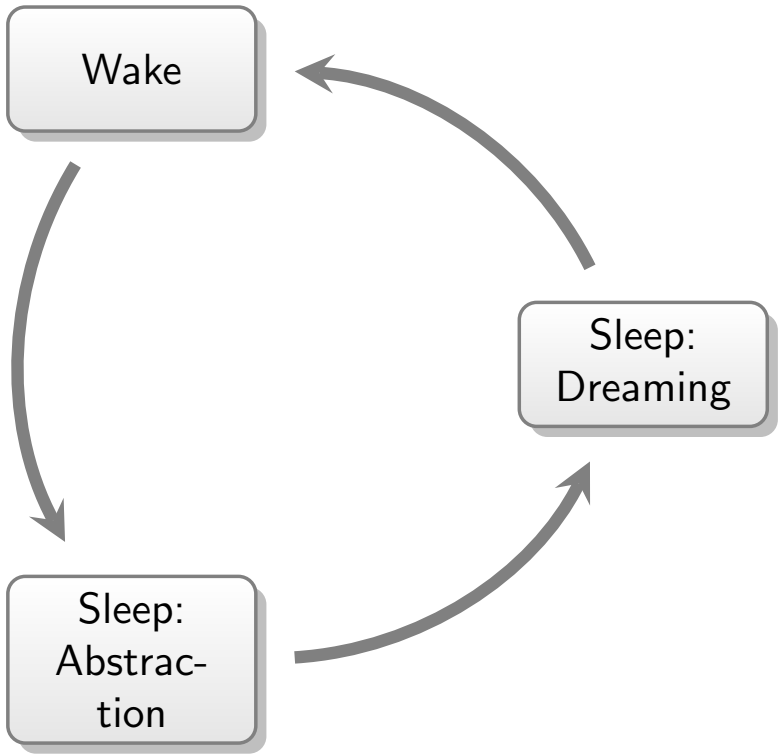
Wake-Sleep Learning



1. **Wake**: find the best program given the current library.
2. **Abstraction**: analyse and compress programs, grow library.
3. **Dreaming**: train a recognition model to predict programs.

m13610190@cgu.edu.tw

Wake-Sleep Learning



1. **Wake**: find the best program given the current library.
2. **Abstraction**: analyse and compress programs, grow library.
3. **Dreaming**: train a recognition model to predict programs.

m13610190@cgu.edu.tw

Learning Example

- Starting with the initial primitives, the model builds a library of more advanced functions (filter, maximum, nth largest).
- Final solution is just five function calls, found in 10 mins and reads naturally vs. 32 function calls in the basis which would require 1072 years.

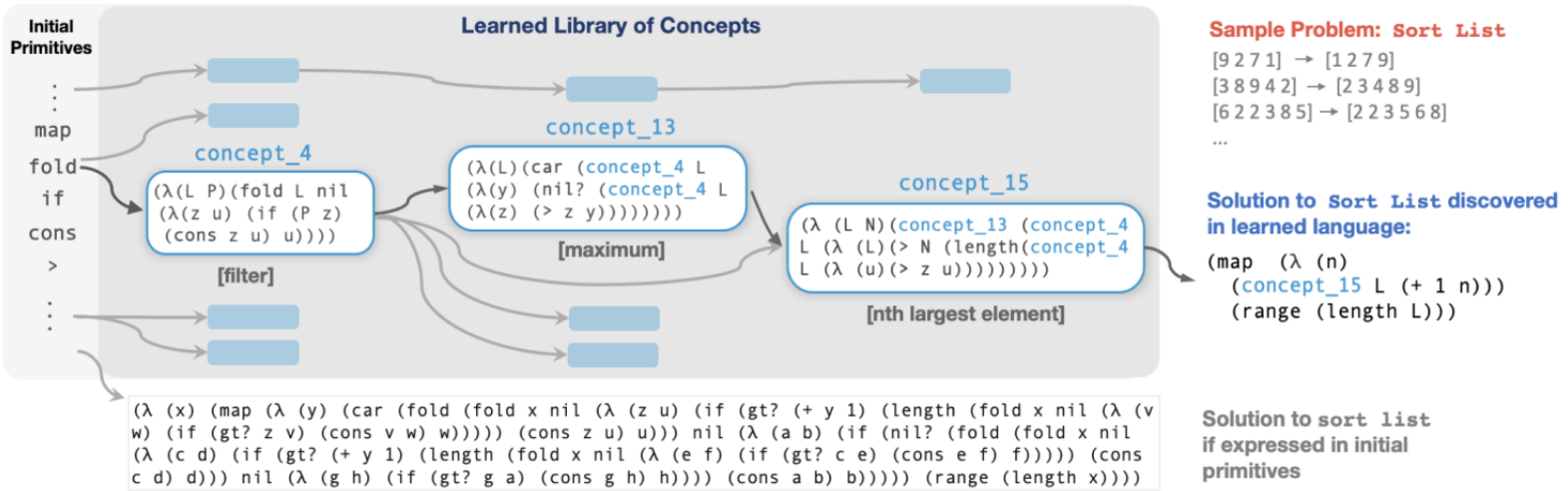


Figure: Learning to sort a list.

m13610190@cgu.edu.tw

Key Results

- Applied to SyGuS program synthesis competition
  - Prior to learning: 4% accuracy, search time 235s; after learning 80% with time of 40s.
  - With same compute as SOTA 84% vs. 82%.
- Against other baselines, DreamCoder always solves the most tasks, and generally in the least time.
- Many great qualitative results on creative tasks (Bongard problems, handwritten characters, turtle graphics).
- Learns the laws of physics from simple mathematical concepts.

m13610190@cgu.edu.tw

Ablation Study

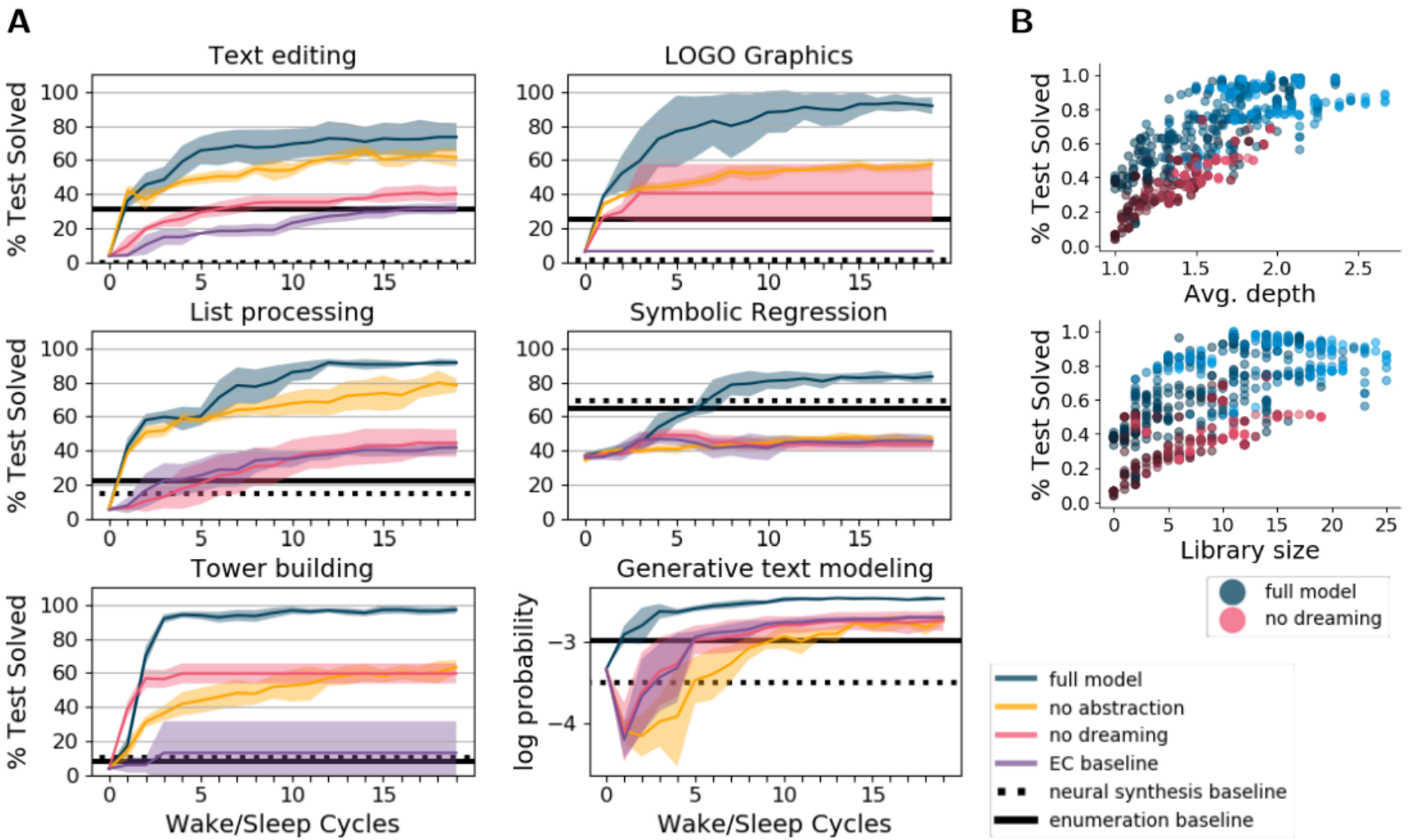


Figure: Ablation studies show that each part of wake/sleep cycle is important.

Conclusion

- Combining generalisation capabilities with a stronger basis presents a powerful idea
- Network architectures could be improved, many symbols exist on graphs
- Can't we just use GenAI? (cf. dev.in) LLMs learn 'hazy' definitions of functions, program synthesis operates on symbols instead of tokens.

m13610190@cgu.edu.tw

Questions?