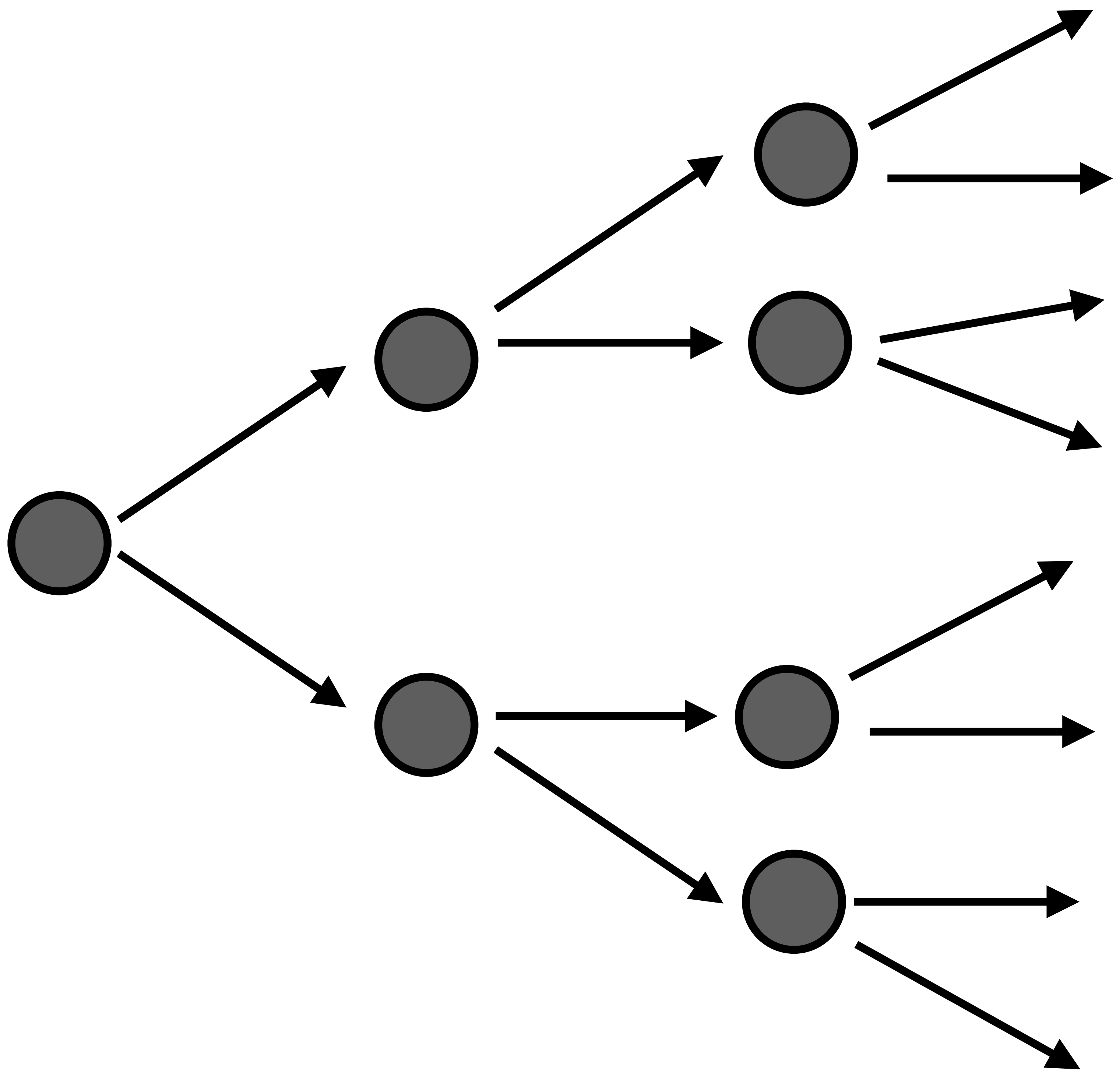
Lecture 24: Inference methods for deep learning

Speaker: Phillip Isola



See also this nice tutorial by Sasha Rush: https://srush.github.io/awesome-o1/o1-tutorial.pdf

Inference

- **Statistics definition:** figuring out properties of a data generating process from samples from that process (in ML we call this "learning" or "training")

- **ML definition:** making predictions about new datapoints using a trained model (in statistics we call this "prediction")

Training

Other names for this:

- *"Statistical inference"*
- *Learning, amortized inference*

Inference

Other names for this:

- *Prediction*
- *Thinking, reasoning, cognition*

Pre-training

Given data, learn a model or representation

Example methods:

- Generative modeling
- Representation learning

Post-training

Given a model and new data, update the model

Example methods:

- Finetuning
- RLHF

Given a model and data during deployment, update the model or its behavior

Example methods:

- Prompting
- In-Context Learning
- Test-Time Training
- Continual learning
- Feedback control

Search

Given a model and a query, find the best answer to the query

Example methods:

- Best-of-N
- Beam search
- MCTS
- Chain-of-Thought

"Reinforcement learning", STaR, self-instruct, self-play, …

The Bitter Lesson

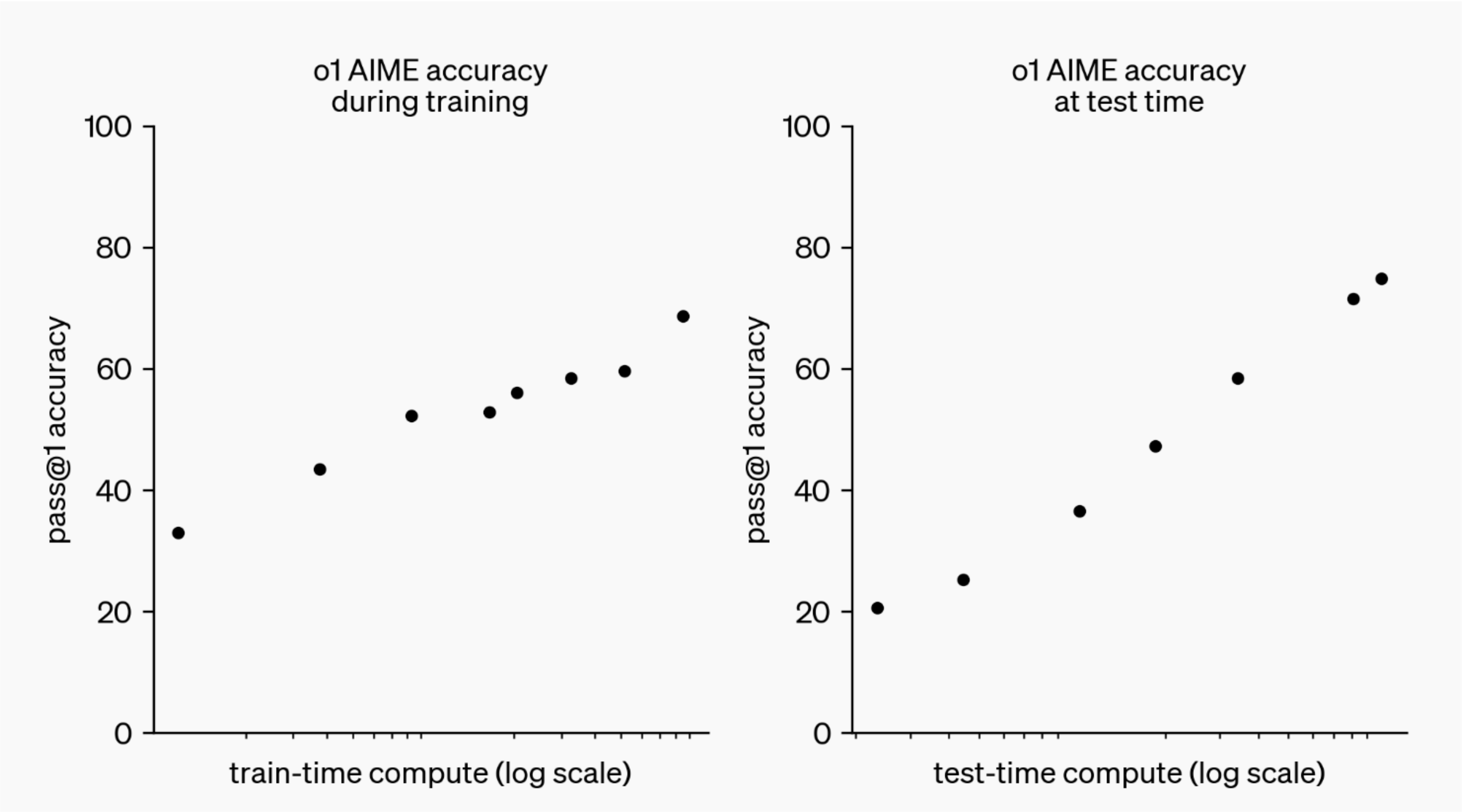
Rich Sutton

March 13, 2019

One thing that should be learned from the bitter lesson is the great power of general purpose methods, of methods that continue to scale with increased computation even as the available computation becomes very great. The two methods that seem to scale arbitrarily in this way are *search* and *learning*.

Training

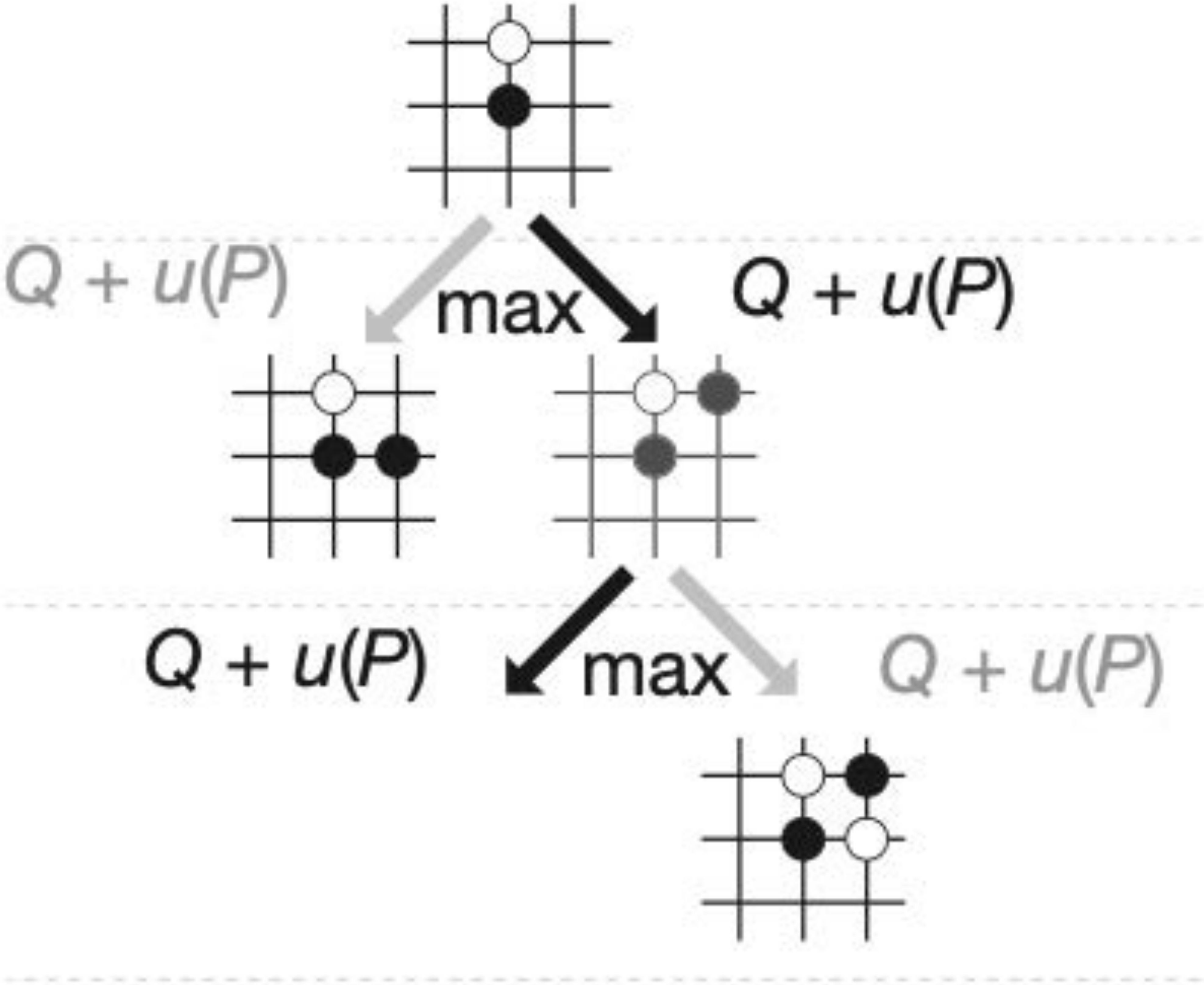
Search



[OpenAI's **o1**: https://openai.com/index/learning-to-reason-with-llms/]

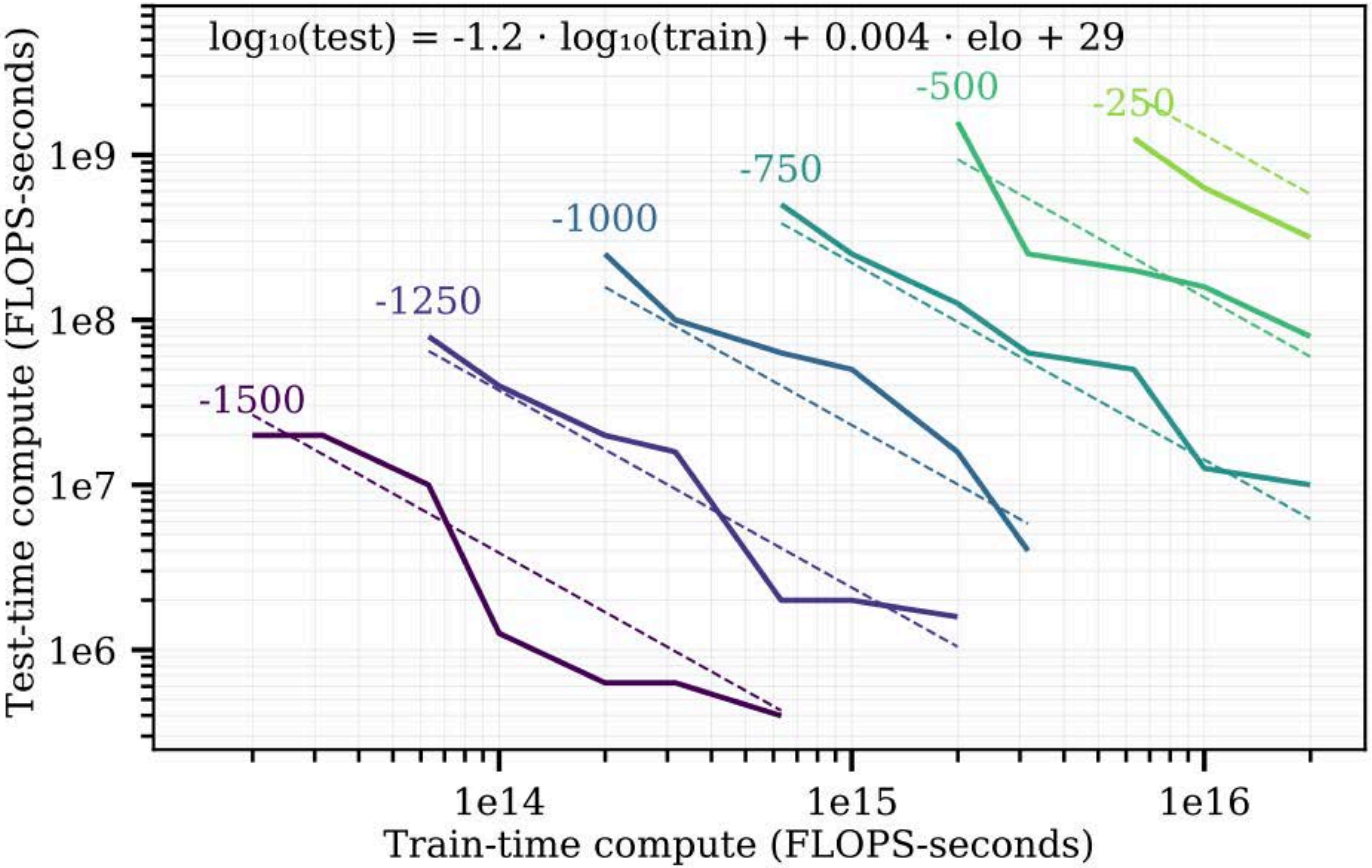
What is search?

Kasparov vs. IBM Deep Blue



[AlphaGo: Silver*, Huang* et al, Nature 2016]

Scaling Scaling Laws with Board Games



[Jones 2021]

Training

Other names for this:
- *"Statistical inference"*
- *Learning, amortized inference*

Inference

Other names for this:
- *Prediction*
- *Thinking, reasoning, cognition*

Pre-training

Given data, learn a model or representation

Example methods:
- Generative modeling
- Representation learning

Post-training

Given a model and new data, update the model

Example methods:
- Finetuning
- RLHF

Given a model and data during deployment, *update the model or its behavior*

Example methods:
- Prompting
- In-Context Learning
- Test-Time Training
- Continual learning
- Feedback control

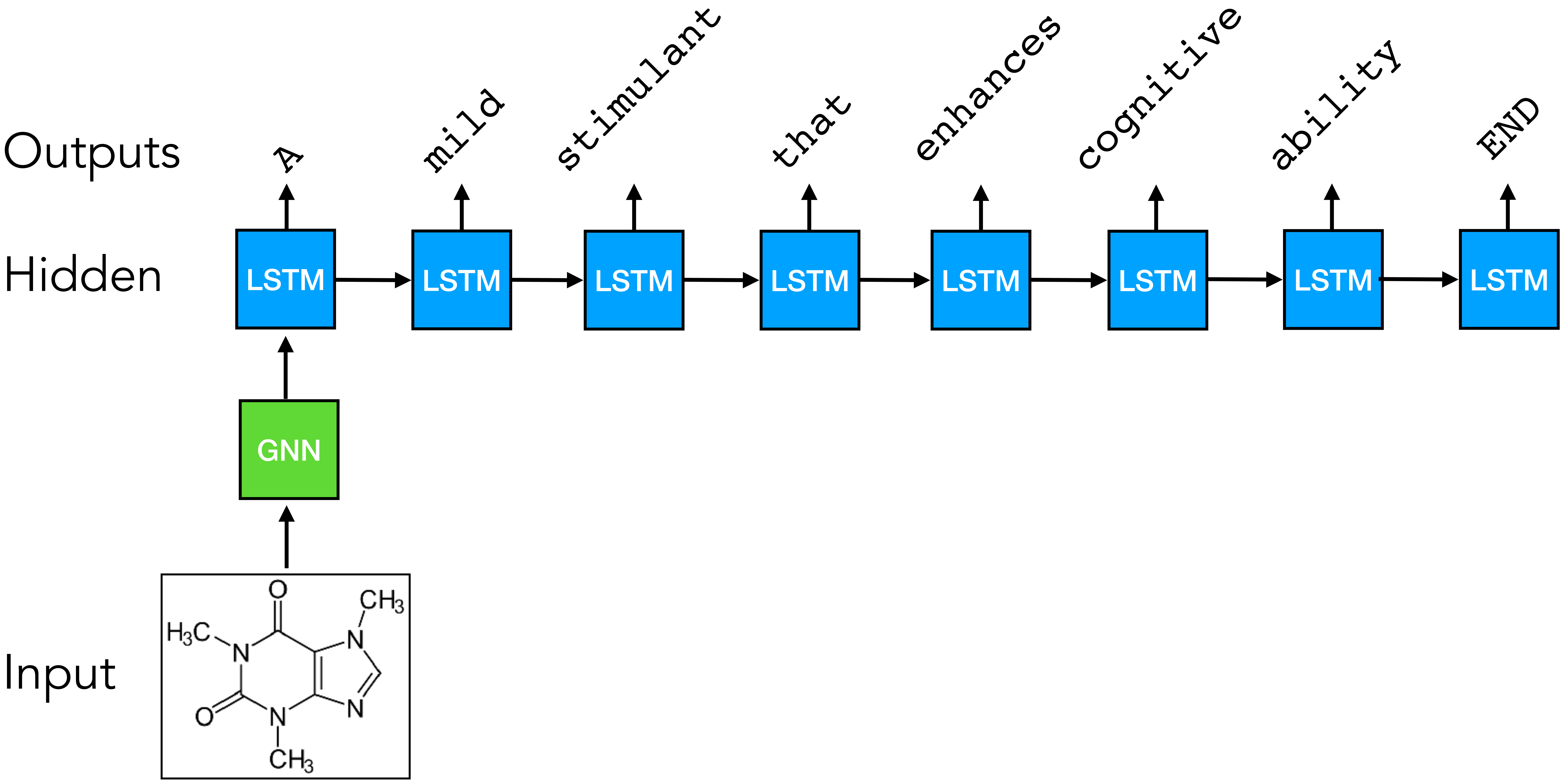
Search

Given a model and a query, find the best answer to the query

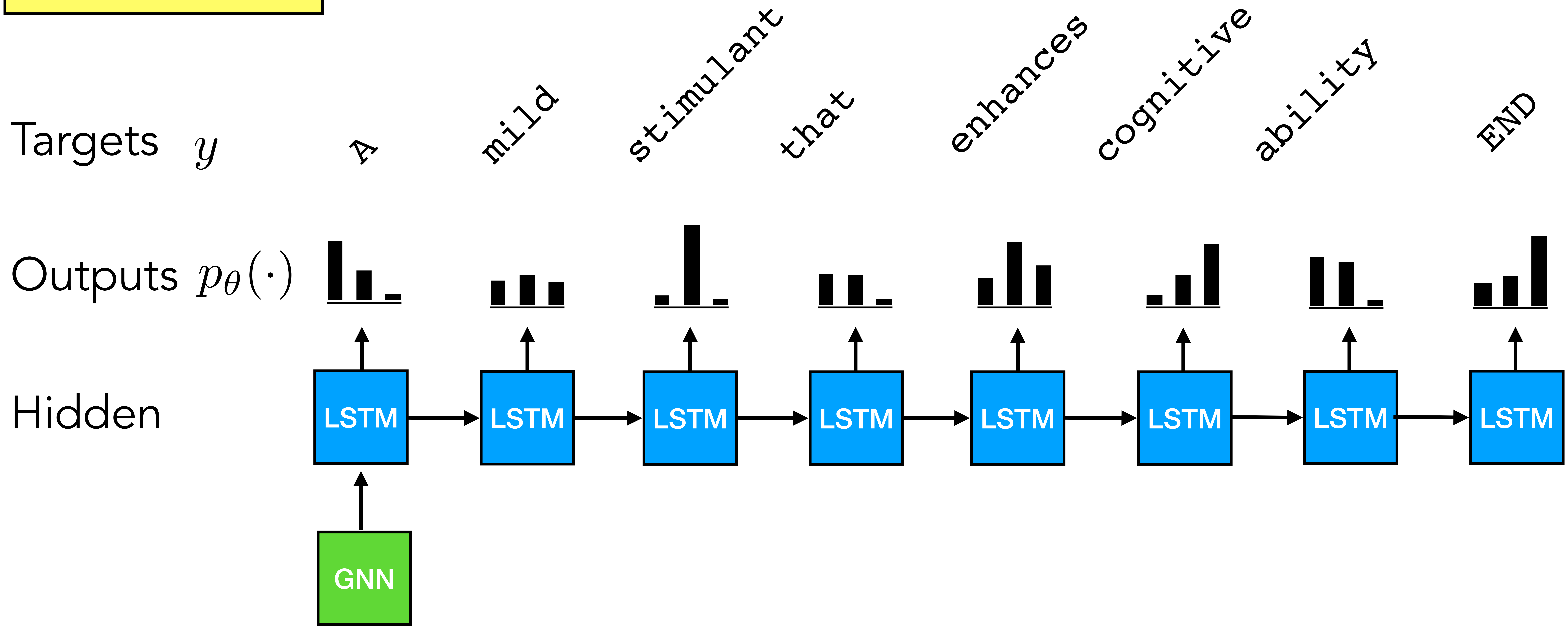
Example methods:
- **Best-of-N**
- **Beam search**
- MCTS
- **Chain-of-Thought**

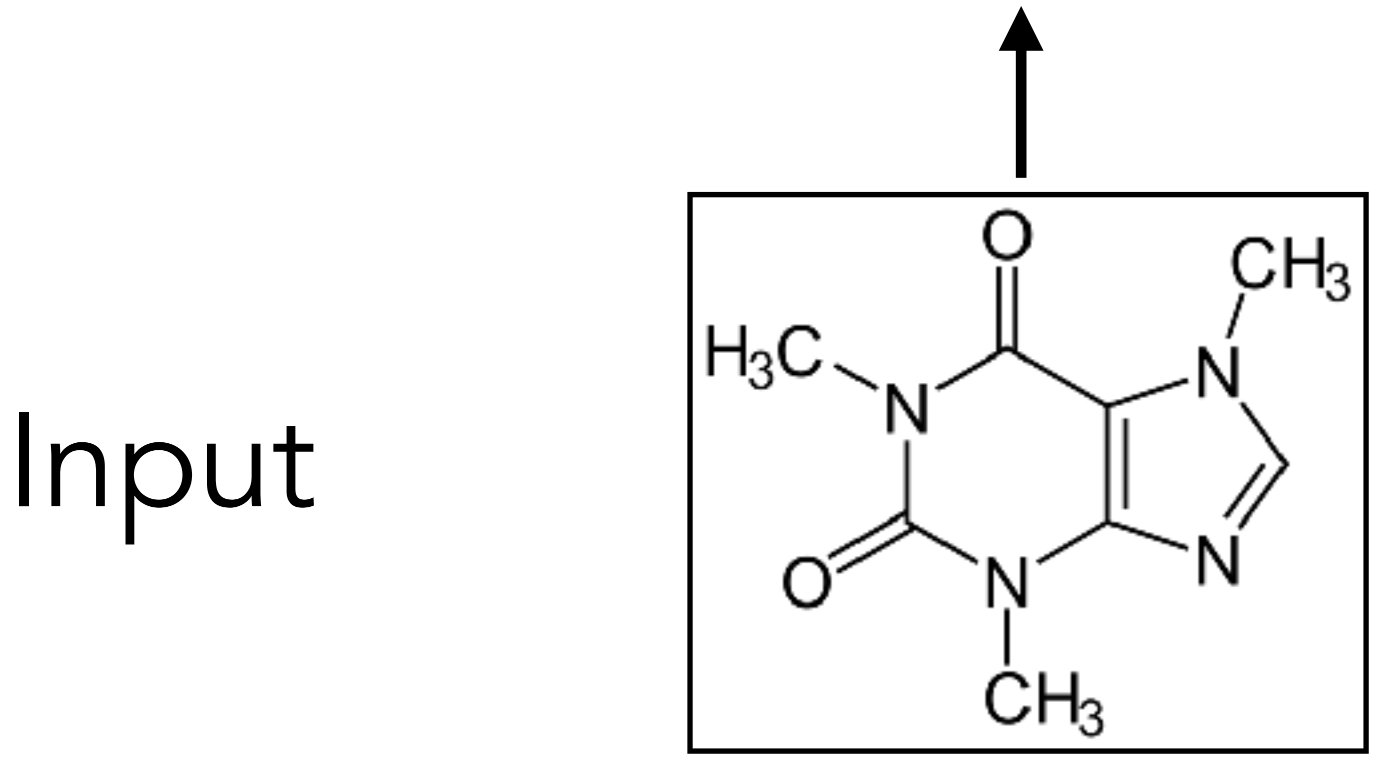
"Reinforcement learning", STaR, self-instruct, self-play, …

Search through an autoregressive model



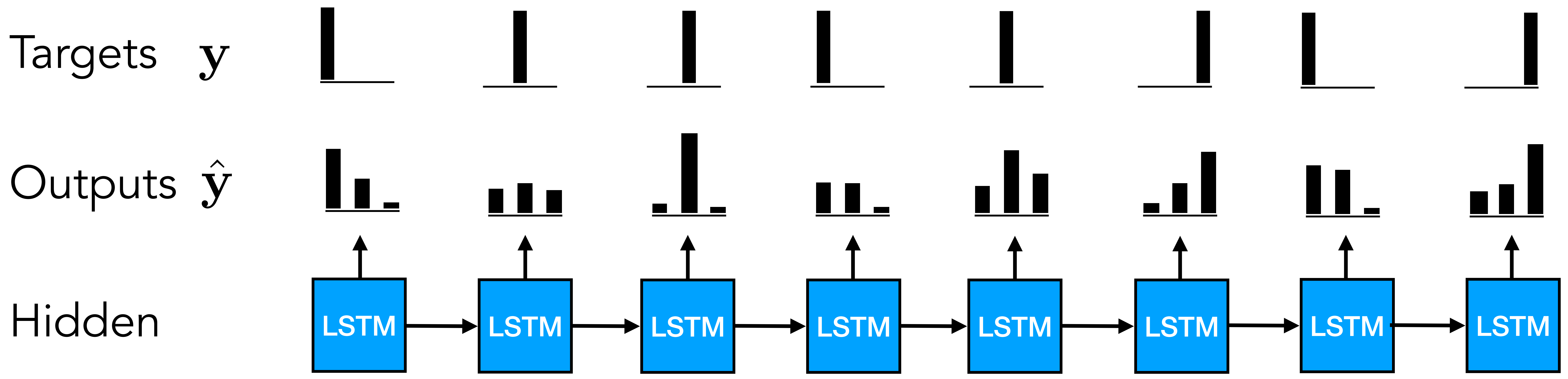
Training

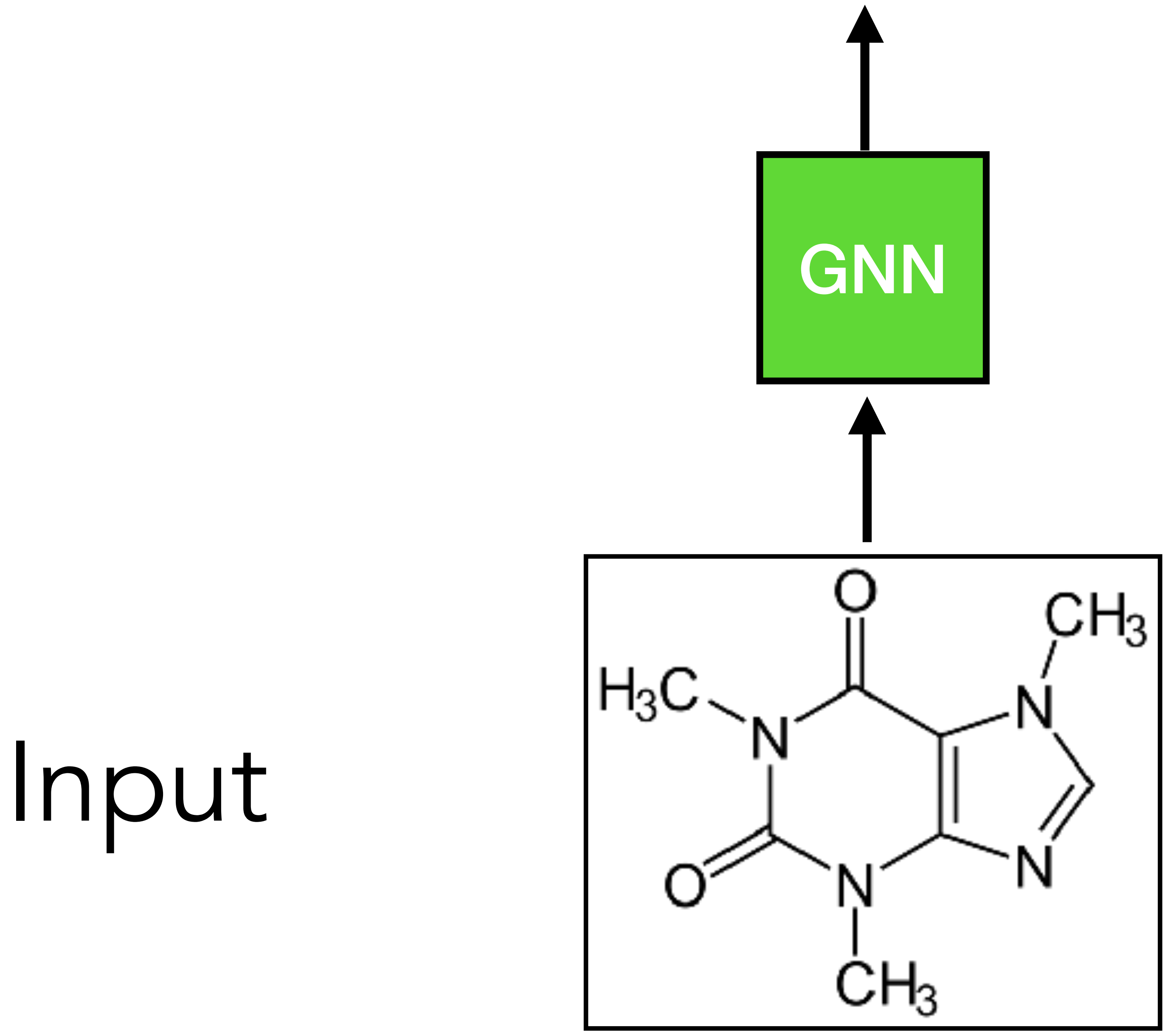




Max-likelihood objective: maximize probability the model assigns to each target word: $\arg\max_\theta \log p_\theta(y)$

Training



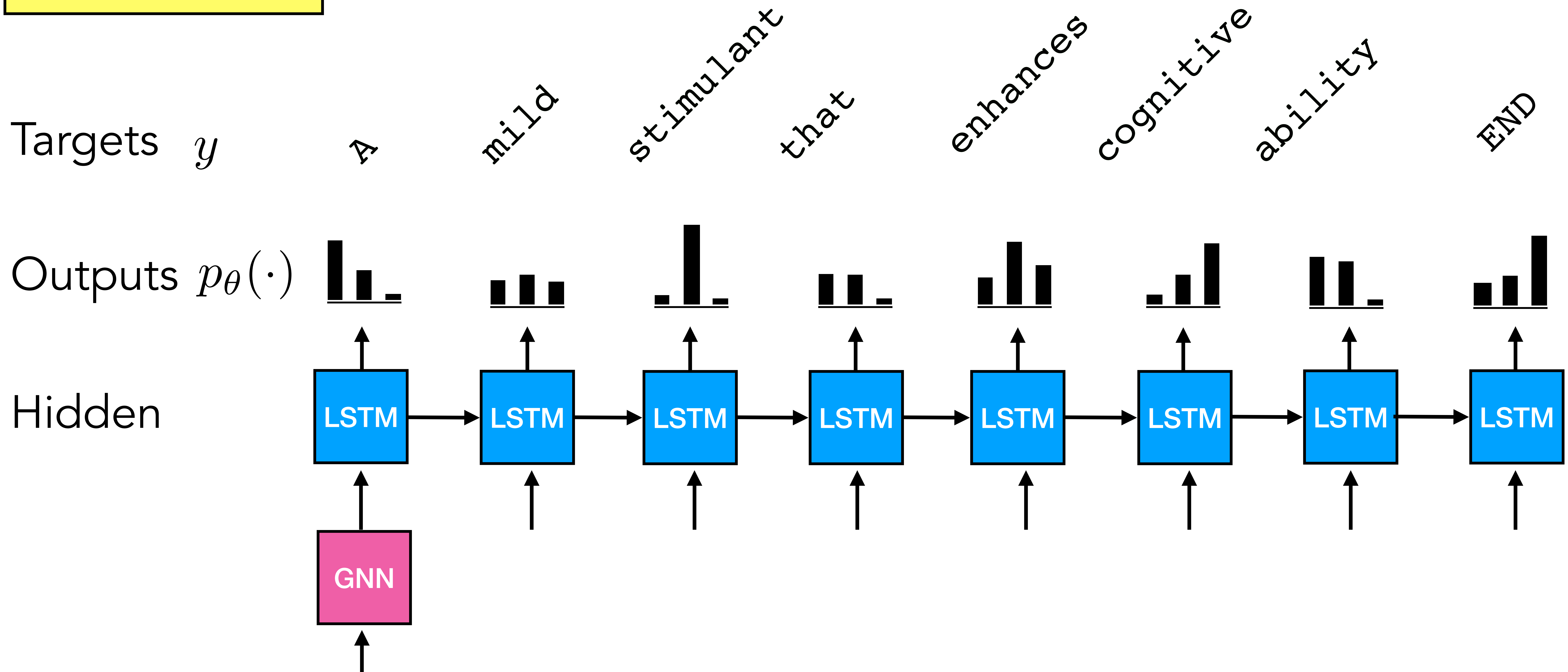


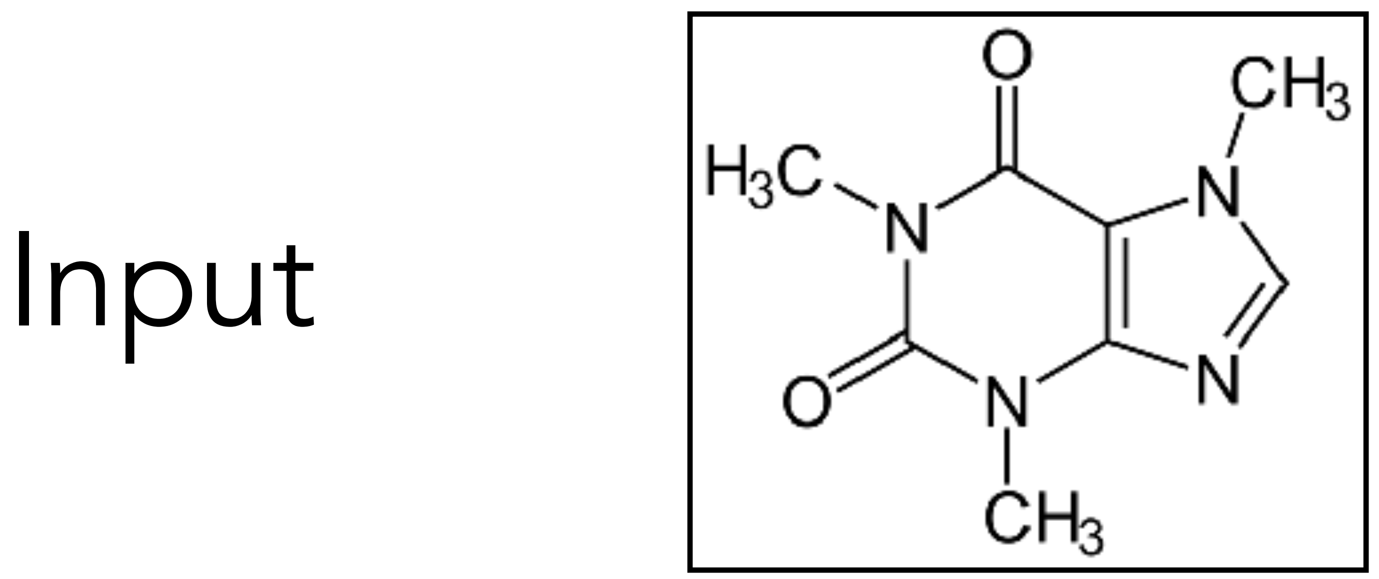
Max-likelihood objective:
minimize cross-entropy between model outputs and one-hot encoded targets.

$$f^* = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} H(\mathbf{y}_i, \hat{\mathbf{y}}_i)$$

Training

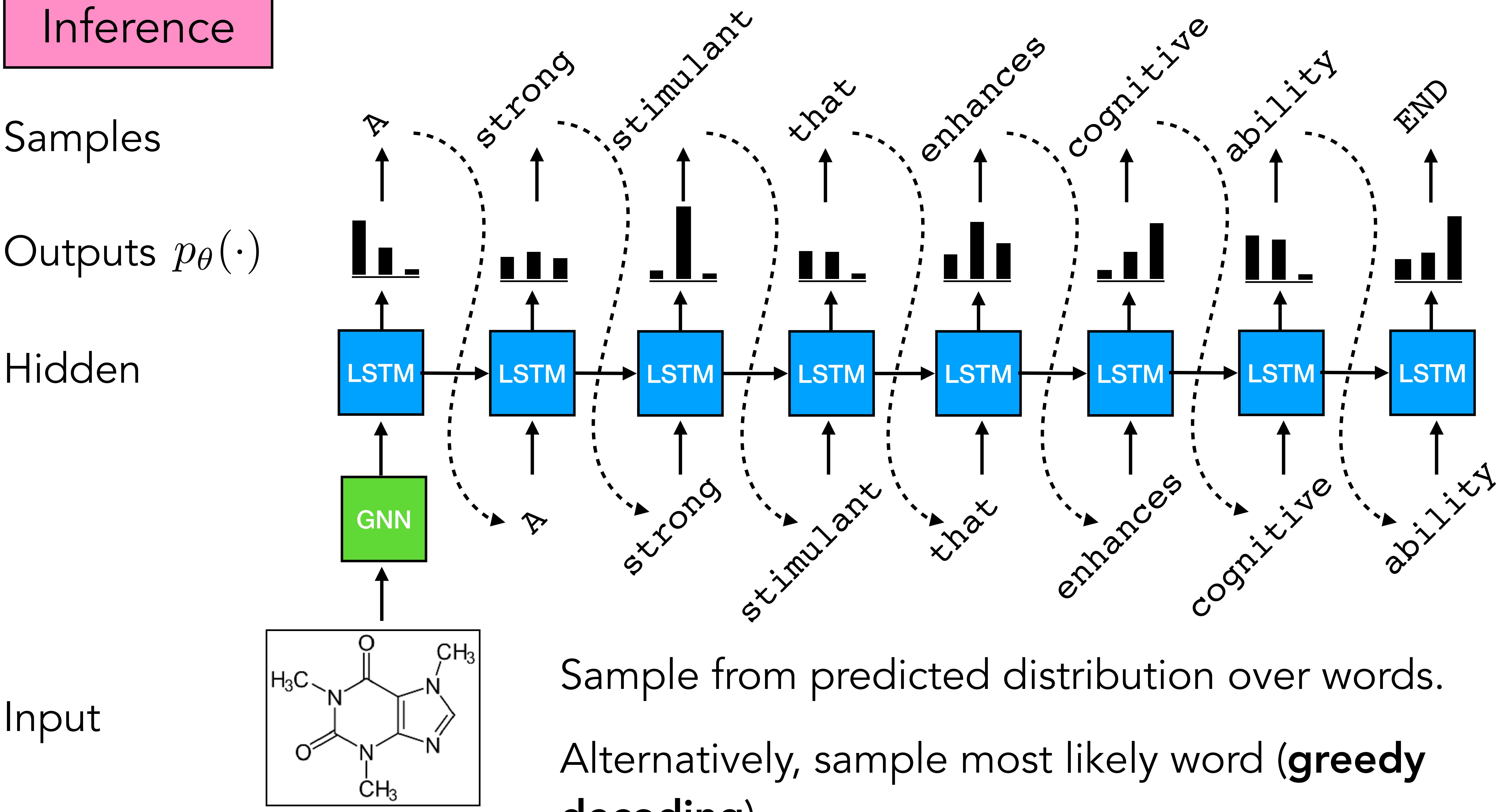
Teacher forcing





Condition each next word prediction on the **ground-truth** preceding word.

Inference



Sample from predicted distribution over words.

Alternatively, sample most likely word (**greedy decoding**).

Greedy sampling does not maximize likelihood!

$$\overbrace{p(\mathbf{X})}^{\{\mathbf{x}_1^*, \dots, \mathbf{x}_n^*\} = \arg\max_{\mathbf{X}} p(\mathbf{X})} = \overbrace{p(\mathbf{x}_n | \mathbf{x}_1, \dots, \mathbf{x}_{n-1})}^{\mathbf{x}_n^* = \arg\max_{\mathbf{x}_n}} \cdots \overbrace{p(\mathbf{x}_2 | \mathbf{x}_1)}^{\mathbf{x}_2^* = \arg\max_{\mathbf{x}_2}} \overbrace{p(\mathbf{x_1})}^{\mathbf{x}_1^* = \arg\max_{\mathbf{x}_1}}$$

(What about for VAEs? What about for diffusion models?)

Inference

Best-of-N

Sample N sequences iid and pick the sequence with highest likelihood:

$$p_\theta(\mathbf{y}_1, \ldots, \mathbf{y}_T | \mathbf{x}) = \prod_i p_\theta(\mathbf{y}_i | \mathbf{y}_1, \ldots, \mathbf{y}_{i-1}, \mathbf{x})$$

`A stimulant that enhances cognitive function.` → p_θ → 0.7

`A stimulant that improves thinking.` → p_θ → 0.5

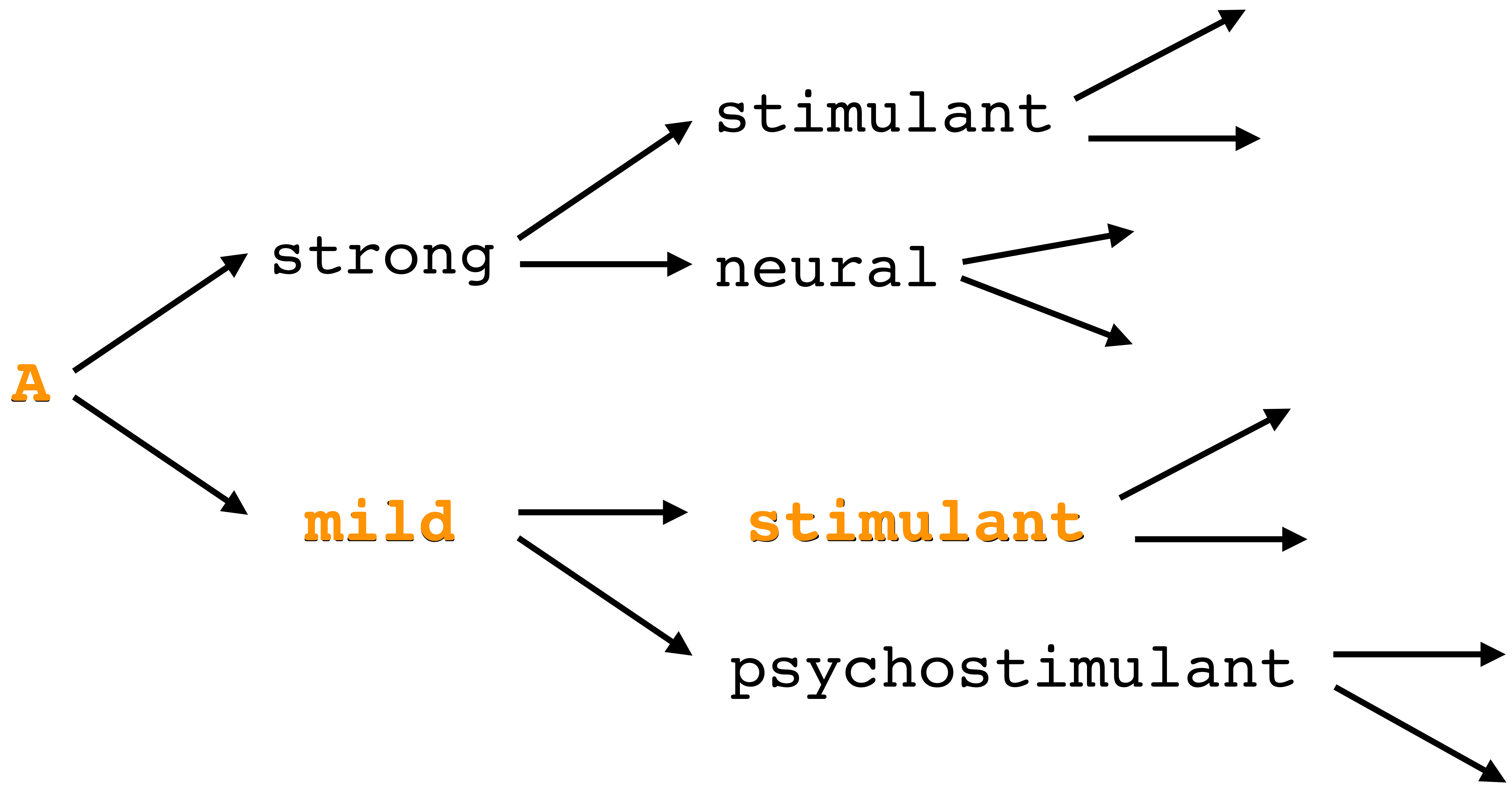
$\boxed{\texttt{A chemical that makes you feel more awake.} \rightarrow p_\theta \rightarrow 0.8}$

…

Inference

Beam search (a form of tree search)

Tree of samples



Sample multiple sequences (top-k greedy completions on each step), then pick the sequence with highest likelihood.

Inference

Chain-of-thought (emergent/learned search)

[Wei et al. 2022]

I want you to use beam search to solve the following problem. Write out your tree of samples then pick the best. Here's the problem:
What is 4*5+10

→ LLM →

...

Final Tree:

markdown Copy code

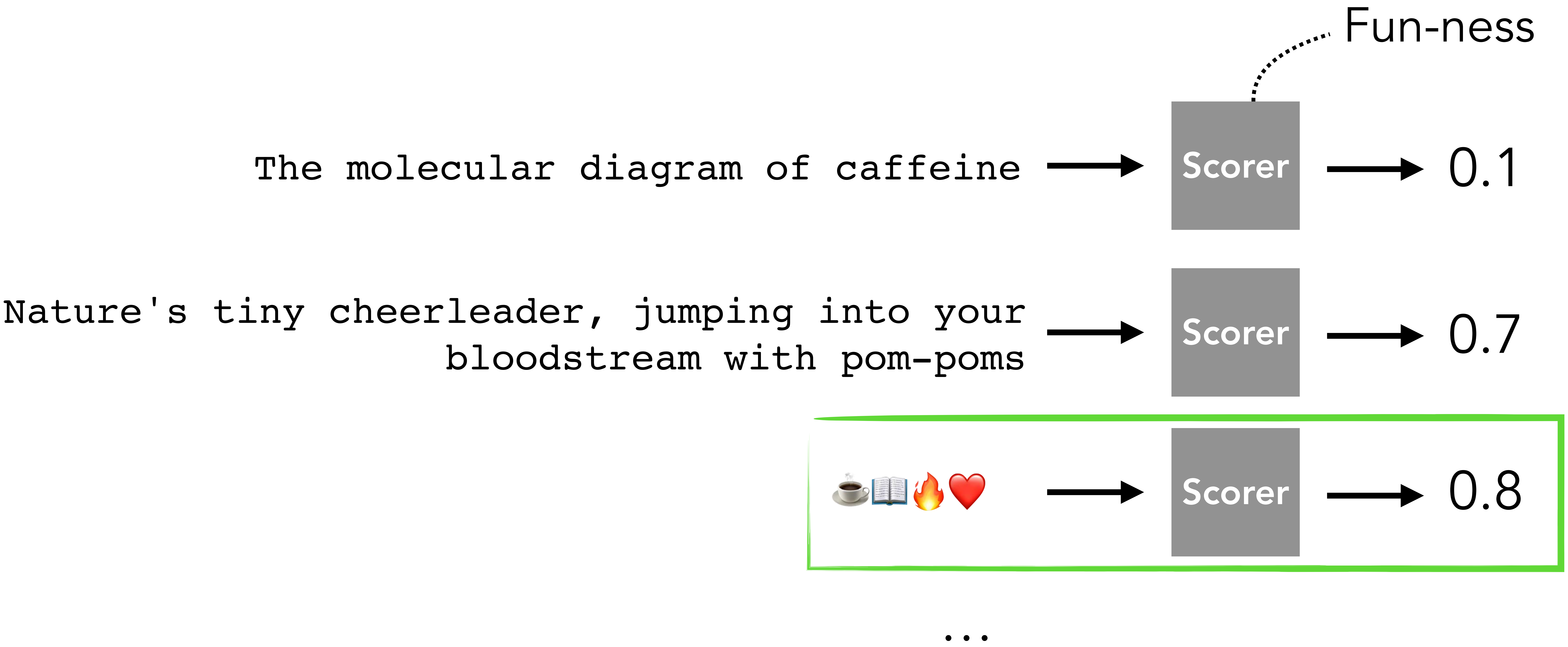
```markdown
          4 * 5 + 10
         /          \
   20 + 10         14 * 5
      |               |
     30              70
```

Conclusion:

Using beam search, the best solution is 30, which comes from correctly following $4 \times 5 + 10 = 30$.

Inference

You can search for samples that maximize other scoring functions, beyond likelihood!



Inference

You can search for samples that maximize other scoring functions, beyond likelihood!

Scientific accuracy

`The molecular diagram of caffeine` → Scorer → 1.0

`Nature's tiny cheerleader, jumping into your bloodstream with pom-poms` → Scorer → 0.3

☕📖🔥❤️ → Scorer → 0.1

...

Inference

Verification

"Write me python code to compute the fibonacci sequence" → LLM

```python
def fibonacci_3(n):
    a, b = 1, 1  # Incorrect initialization
    for _ in range(n):
        a = b
        b = a + b
    return a  # Returns an incorrect Fibonacci value
```

→ Verifier → incorrect

```python
def fibonacci_2(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci_2(n - 1) + fibonacci_2(n - 2)
```

→ Verifier → correct

Key idea: *verification is easier than generation.*

Steering model outputs toward human preferences

Great objectives for AI systems

1. Given lots of wonderful data, imitate it (supervised learning, SSL, generative models, etc)

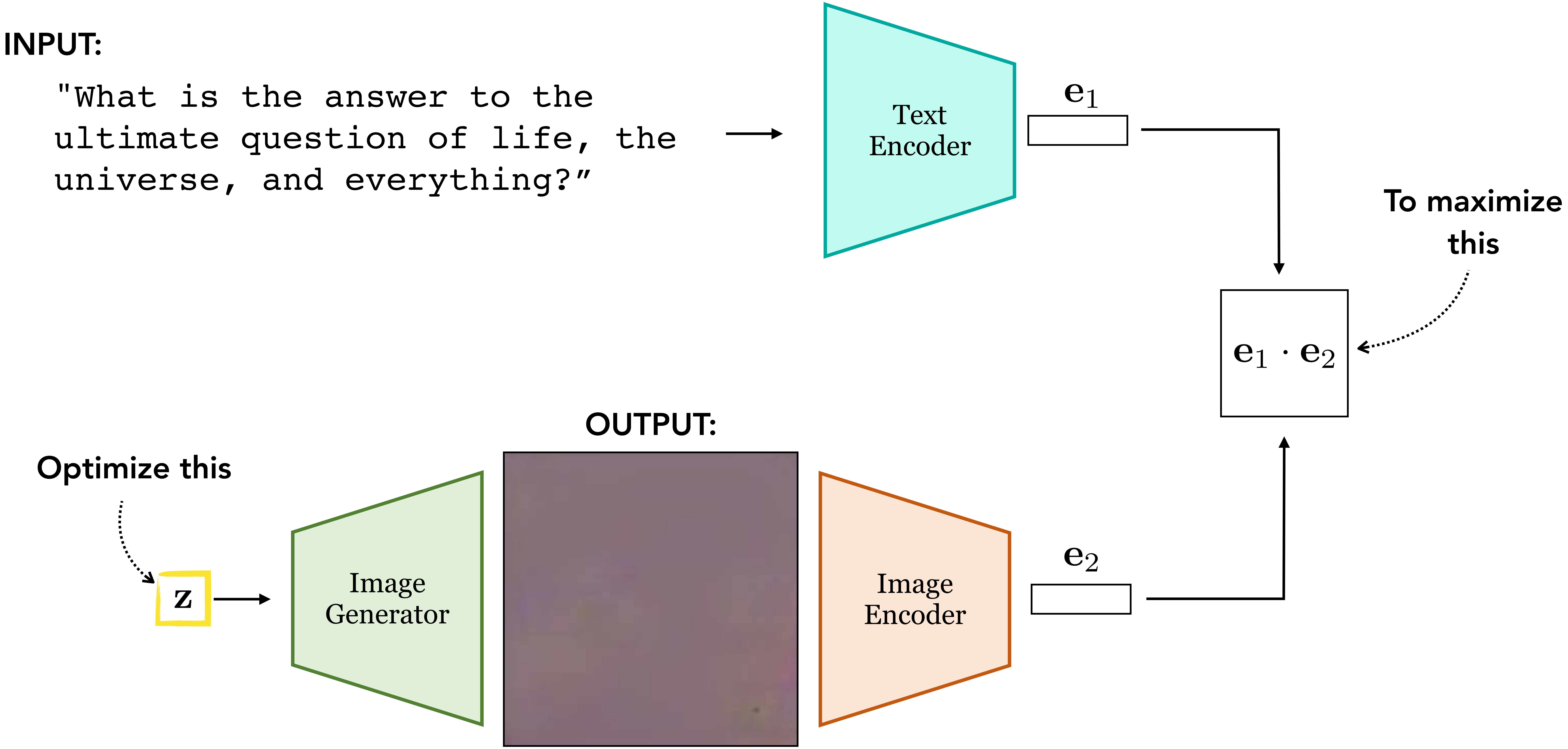
2. …

Other than "imitate wonderful data", what might be other good, general-purpose objectives

Make imagery that people find meaningful

memorable/aesthetic/evocative/…

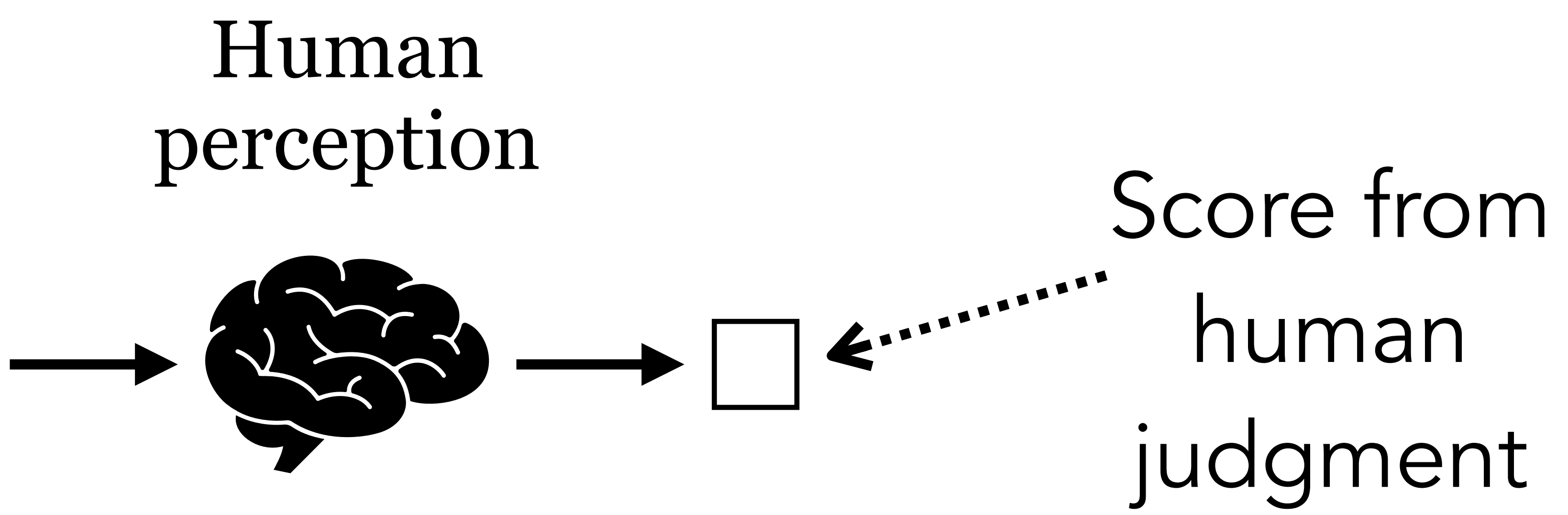
Generating data that optimizes a scoring function

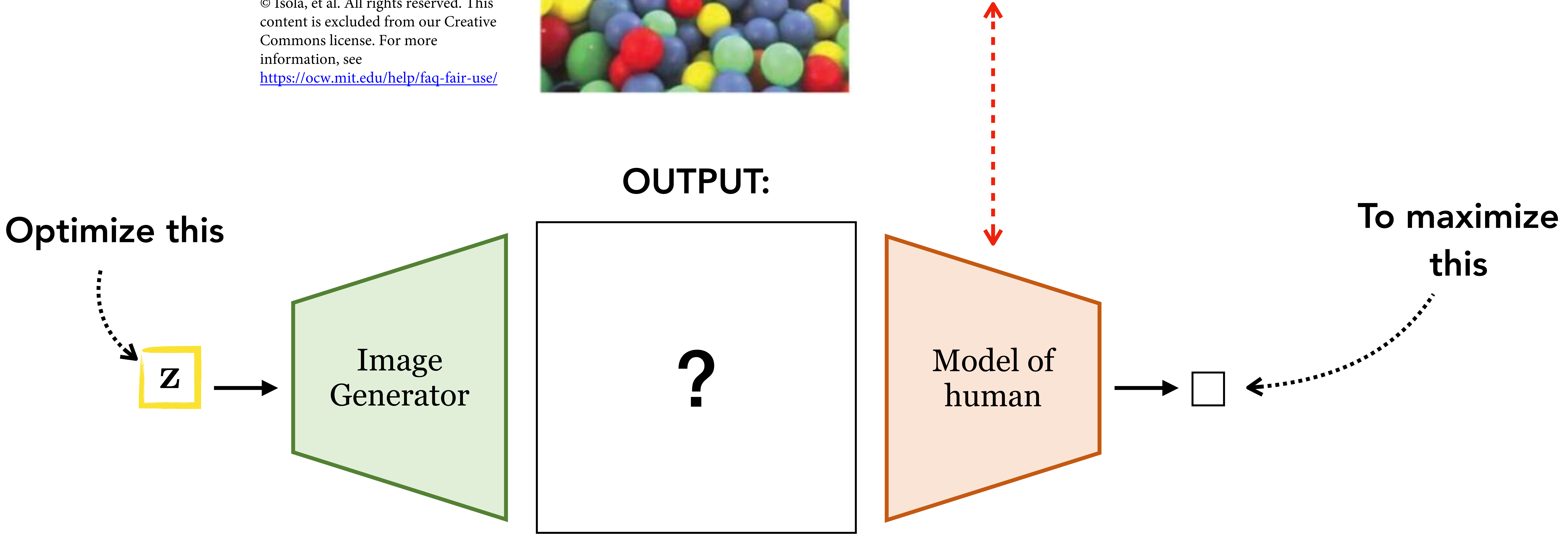


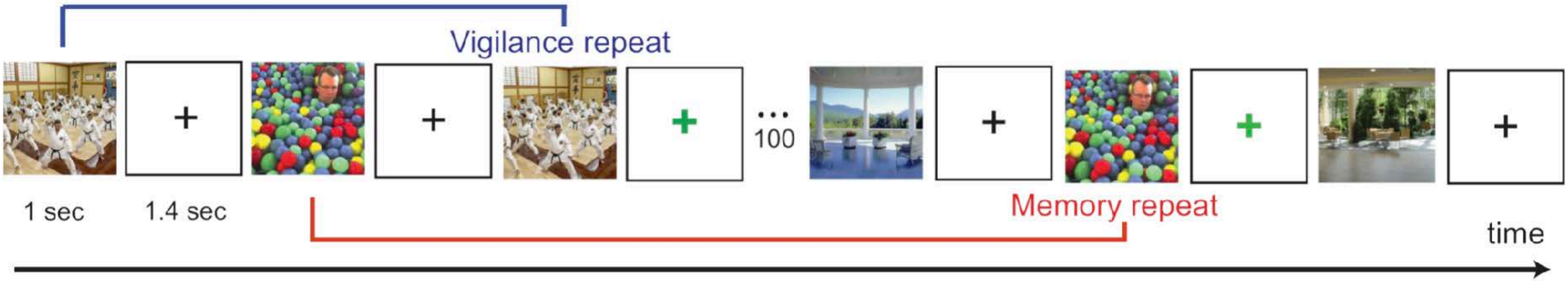
[e.g., CLIP + VQGAN]

Code: https://colab.research.google.com/drive/1_4PQqzM_0KKytCzWtn-ZPi4cCa5bwK2F?usp=sharing

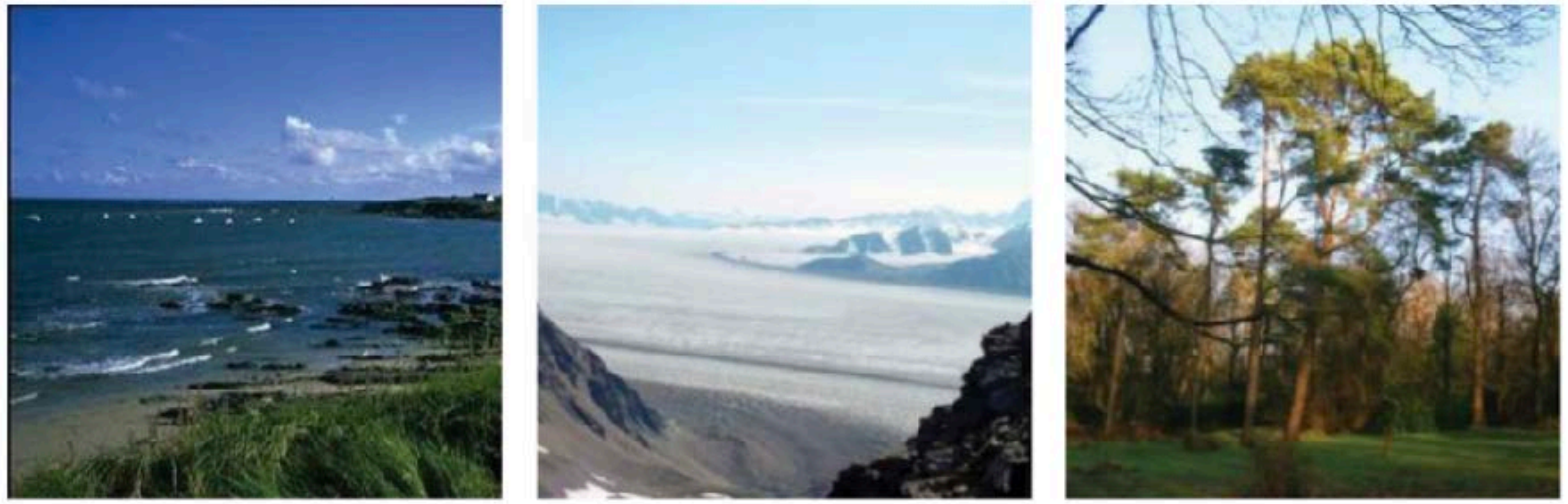
Generating data that optimizes human perception







Forgettable



Memorable

[Isola et al. 2011, Khosla et al. 2015]

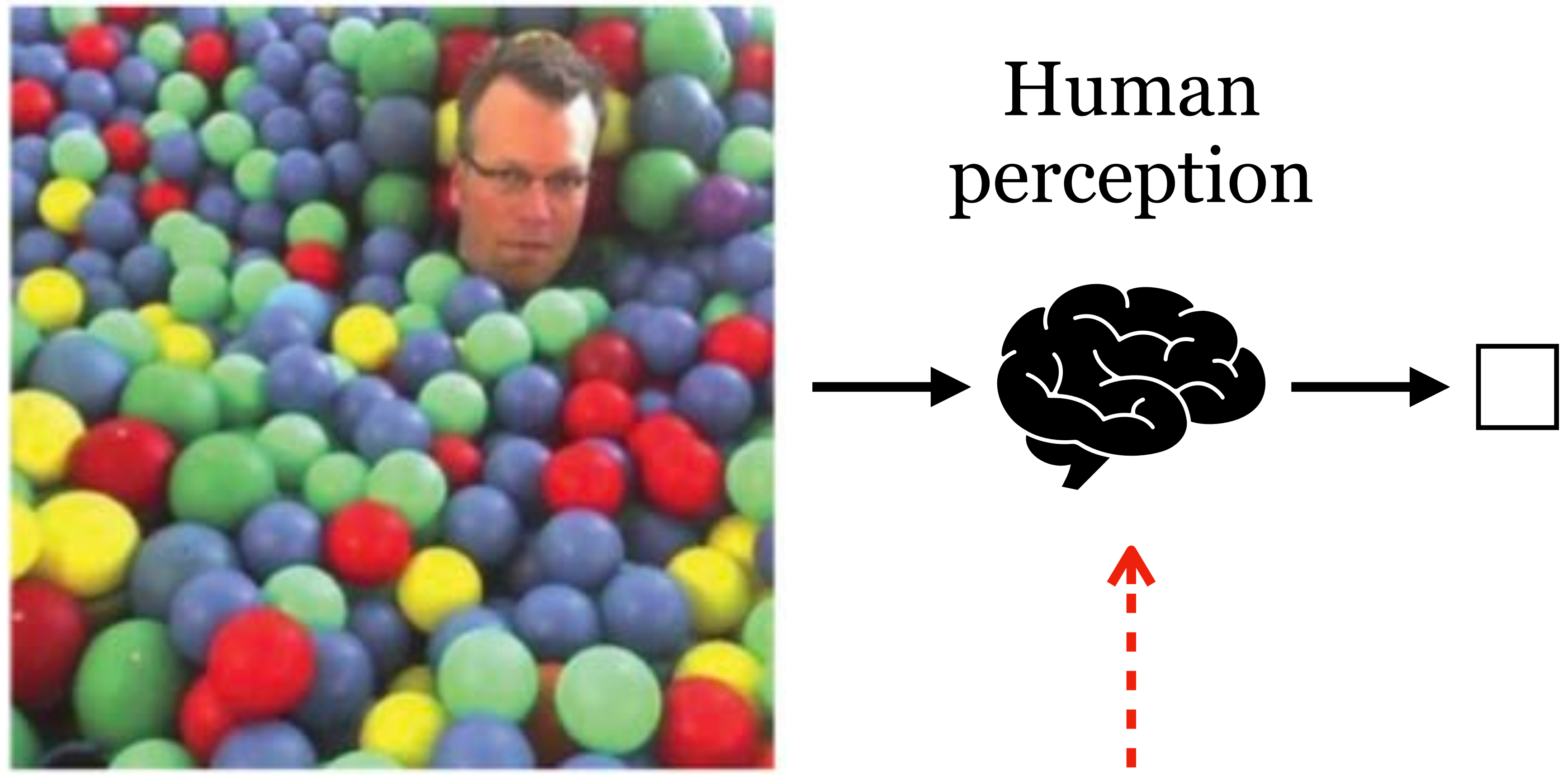
GANalyze: Toward Visual Definitions of Cognitive Image Properties

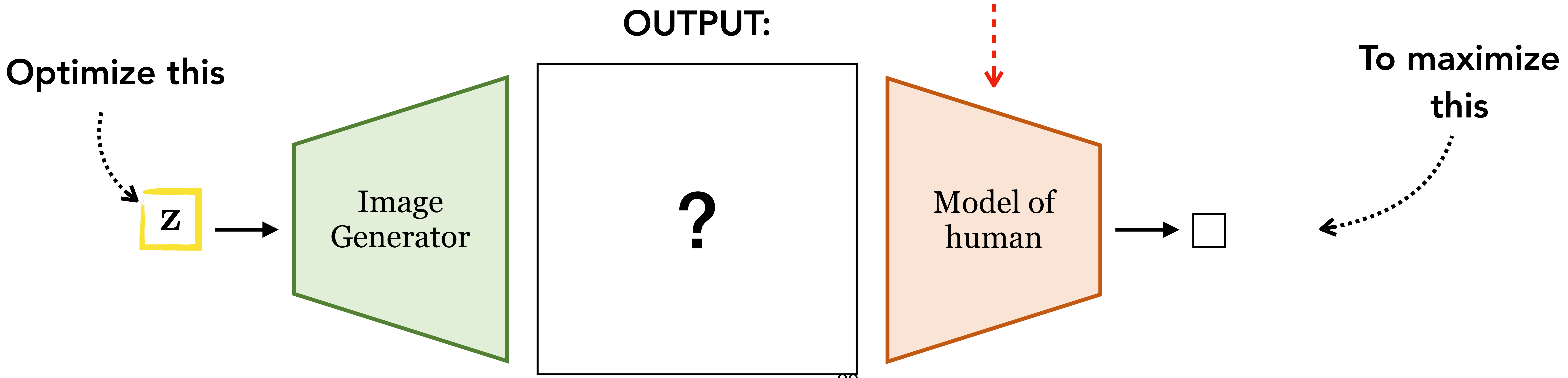
Lore Goetschalkx*, Alex Andonian*, Aude Oliva, Phillip Isola

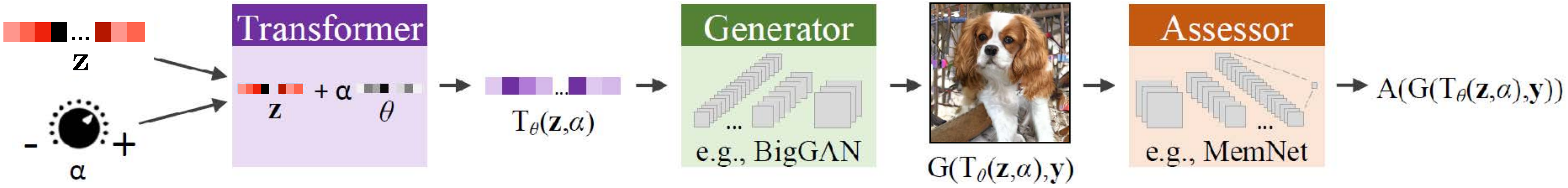
ICCV 2019

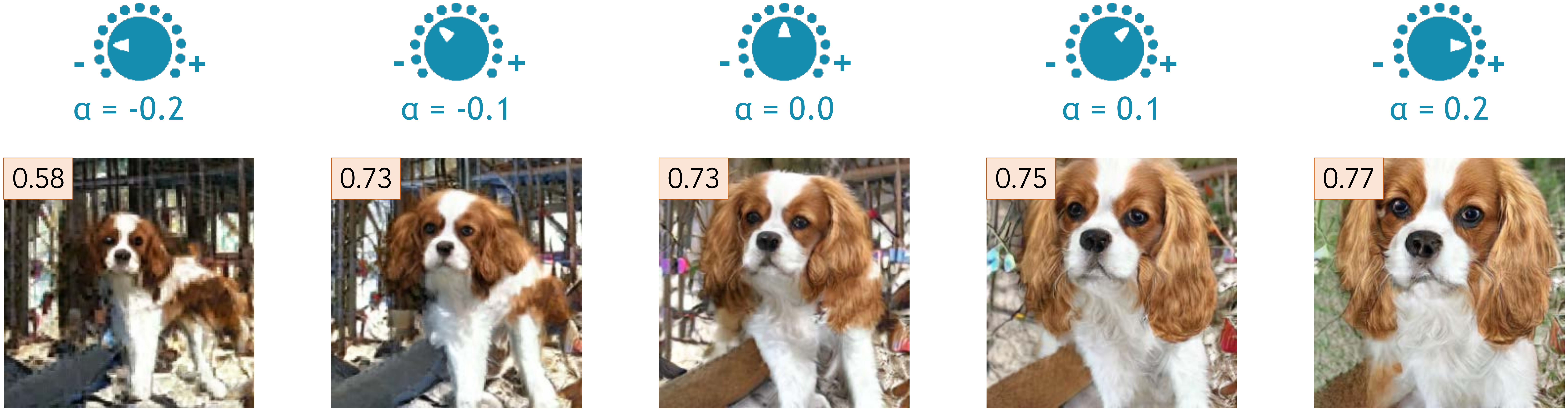
Lore Goetschalckx

Alex Andonian

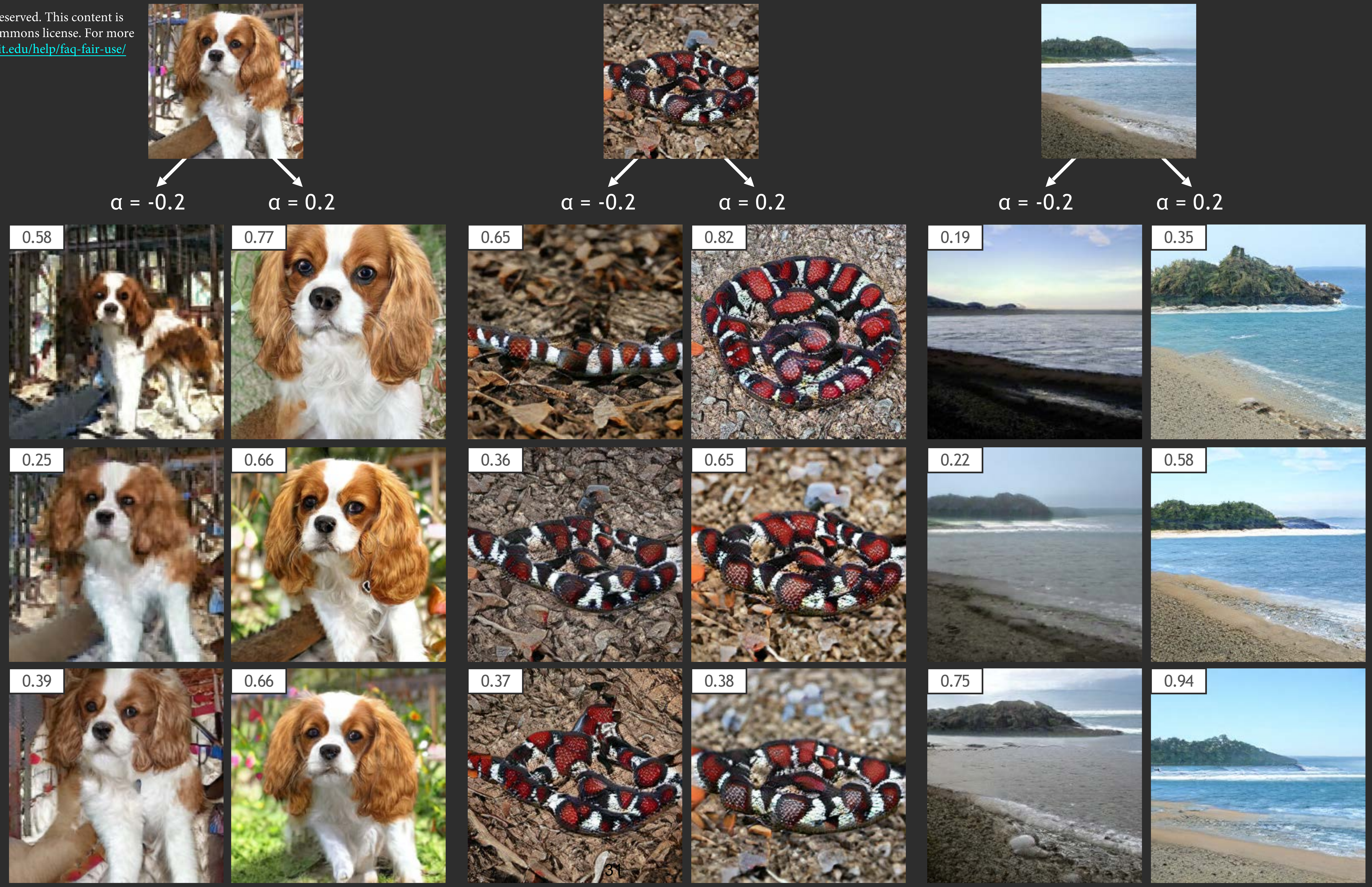








Probe into human perception


α = -0.2
α = 0.2
α = -0.2
α = 0.2
α = -0.2
α = 0.2
Memorability
0.58
0.77
0.65
0.82
0.19
0.35
Aesthetics
0.25
0.66
0.36
0.65
0.22
0.58
Emotional Valence
0.39
0.66
0.37
0.38
0.75
0.94

New paradigm: “Just ask”

Chain-of-thought is “just ask for symbolic search”

Can we do “just ask for continuous reward optimization”?

https://evjang.com/2021/10/23/generalization.html

Images by Phillip Isola generated in DallE.

Images by Phillip Isola generated in DallE.

Images by Phillip Isola generated in DallE.

An very very beautiful painting of a mountain next to a waterfall.

Report issue P

Images by Phillip Isola generated in DallE.

Images by Phillip Isola generated in DallE.

Images by Phillip Isola generated in DallE.

"Make it more"


Jon Barron @jon_barron · Nov 29, 2023
Level 2 Reuben


Jon Barron @jon_barron · Nov 29, 2023
Level 1e15 Reuben

"Make it more"


Jon Barron @jon_barron · Nov 29, 2023
Level 1e40 Reuben


Jon Barron @jon_barron · Nov 29, 2023
A Level 1e-40 Reuben. I'm guessing it's trying to to suggest that the sandwich is microscopically small.

Training

Other names for this:

- *"Statistical inference"*
- *Learning, amortized inference*

Inference

Other names for this:

- *Prediction*
- *Thinking, reasoning, cognition*

Pre-training

Given data, learn a model or representation

Example methods:

- Generative modeling
- Representation learning

Post-training

Given a model and new data, update the model

Example methods:

- Finetuning
- RLHF

Given a model and data during deployment, update the model or its behavior

Example methods:

- Prompting
- **In-Context Learning**
- **Test-Time Training**
- Continual learning
- Feedback control

Search

Given a model and a query, find the best answer to the query

Example methods:

- Best-of-N
- Beam search
- MCTS
- Chain-of-Thought

"Reinforcement learning", STaR, self-instruct, self-play, …

In-Context Learning

Review: the whole thing 's fairly lame , making it par for the course for disney sequels . $\boldsymbol{x}^{(1)}$

Answer: Negative $\boldsymbol{y}^{(1)}$

Review: this quiet , introspective and entertaining independent is worth seeking . $\boldsymbol{x}^{(2)}$

Answer: Positive $\boldsymbol{y}^{(2)}$

Review: this quiet , introspective and entertaining independent is worth seeking . $\boldsymbol{x}$

Answer:

↓

LM

↓

"Positive"

$$P(\cdot \mid \boldsymbol{x}^{(1)}, \boldsymbol{y}^{(1)}, \ldots, \boldsymbol{x}^{(N)}, \boldsymbol{y}^{(N)}, \boldsymbol{x})$$

Need to turn labels into natural language

Extra text on top of input/outputs

[Example from Zhao et al. '21]

“Transformers learn in-context by gradient descent”

- Let f_θ be a transformer, x a query and $\mathbf{c}$ the context.
- $f_\theta(x, \mathbf{c}) = f_{\theta'}(x)$
- What’s the relationship between θ and θ' ?
- This paper shows a case where $\theta' = \theta + \lambda \nabla_\theta \mathscr{L}(f_\theta(x), \mathbf{c})$
- That is, in-context learning, in certain cases, can be expressed as gradient descent over the context examples, to update the final mapping from query to answer.

[von Oswald et al. ICML 2023]

- This result raises a tantalizing possibility: instead of doing ICL over $\mathbf{c}$, why not do gradient descent over $\mathbf{c}$?

- This has several possible advantages:

  - ICL is learned learning. It isn't guaranteed to work. And it often doesn't work. Gradient descent is not learned. It is guaranteed to improve performance on the in-context examples.

  - Gradient descent can leverage more test-time compute than ICL. You can run it forever.

- And some possible disadvantages:

  - ICL could be a smarter learning algorithm than gradient descent.

Post-training

Test-Time Training

[Sun et al. ICML 2020]

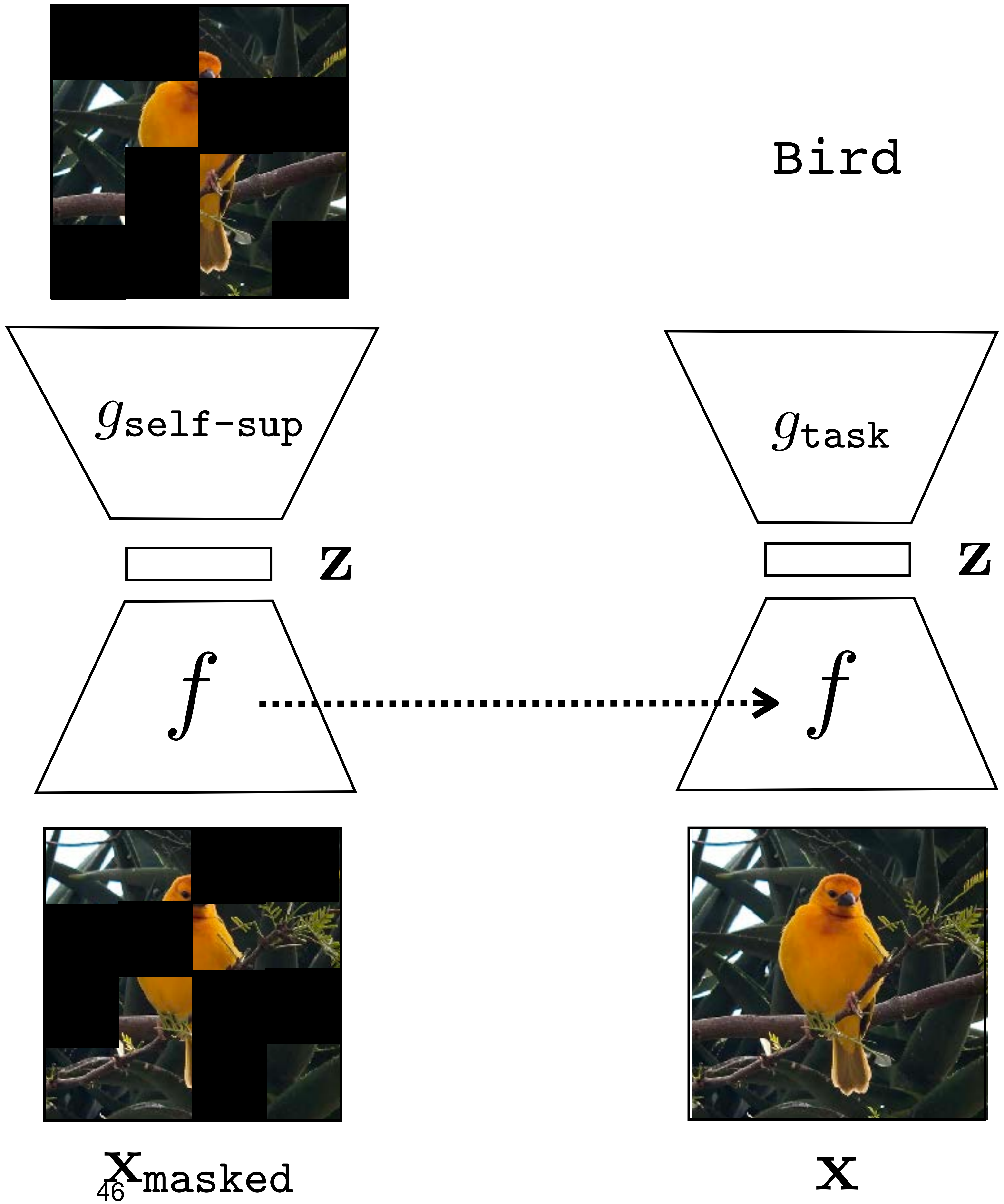
- Update the weights of your model *as a function of the test query (/ queries).*
- If the test query includes few-shot examples, $\{x_1, y_1, x_2, y_2, \ldots, x_q, ?\}$, then this should clearly improve performance on those examples.
- What if the test query is just a x, or a set of unlabeled examples $\{x_1, x_2, \ldots, x_q, ?\}$

Post-training

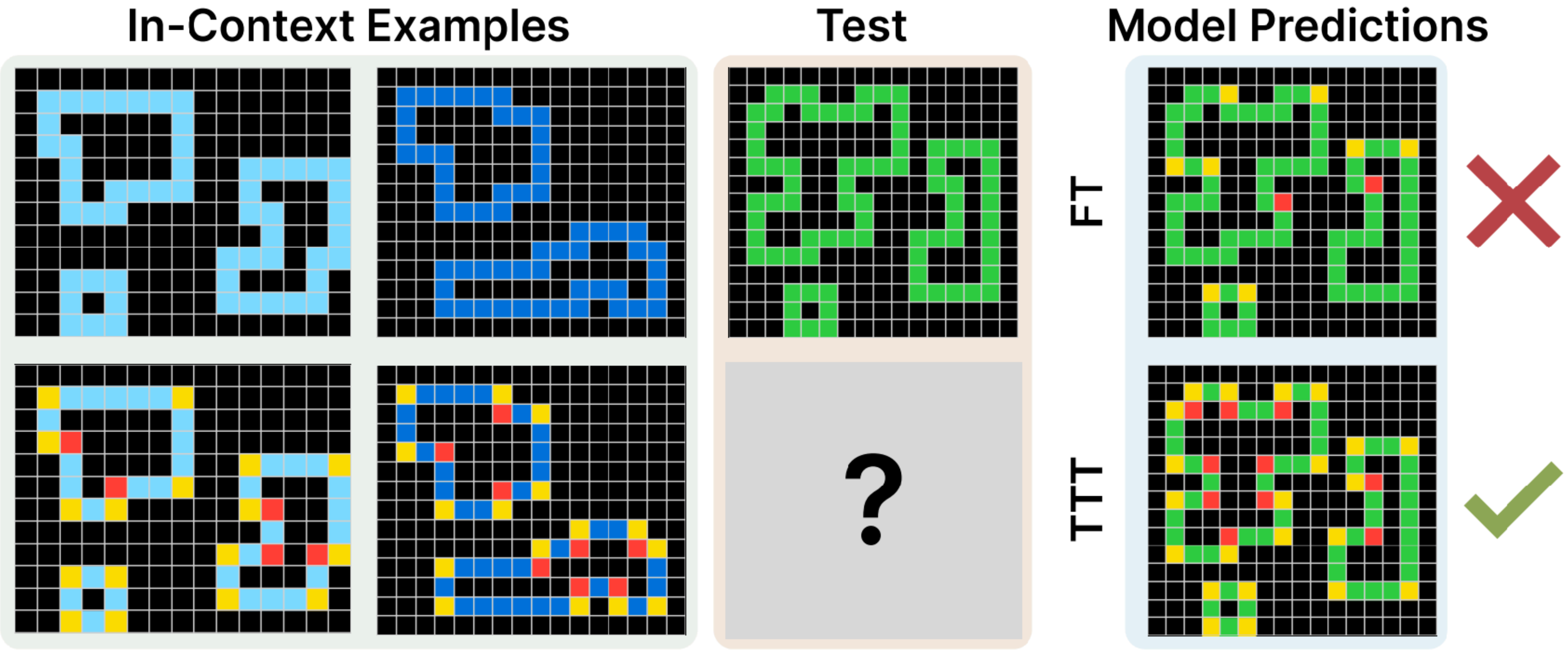
Test-Time Training

[Sun et al. ICML 2020]

Then we can update the weights using test-time self-supervised learning!



"The Surprising Effectiveness of Test-Time Training for Abstract Reasoning"



[Akyürek, Damani, Qiu, Gut, Kim, Andreas, 2024]

"The Surprising Effectiveness of Test-Time Training for Abstract Reasoning"

- 61.9% performance on ARC, which is current SOTA
- Mostly with an "old" method (GD on supervised examples; but examples that are highly relevant to each test query)
- Raises question: do we need fancy new search methods? Or is GD still enough?

[Akyürek, Damani, Qiu, Guo, Kim, Andreas, 2024]

Training

Other names for this:
- *"Statistical inference"*
- *Learning, amortized inference*

Inference

Other names for this:
- *Prediction*
- *Thinking, reasoning, cognition*

Pre-training

Given data, learn a model or representation

Example methods:
- Generative modeling
- Representation learning

Post-training

Given a model and new data, update the model

Example methods:
- Finetuning
- RLHF

Given a model and data during deployment, update the model or its behavior

Example methods:
- Prompting
- In-Context Learning
- Test-Time Training
- Continual learning
- Feedback control

Search

Given a model and a query, find the best answer to the query

Example methods:
- Best-of-N
- Beam search
- MCTS
- Chain-of-Thought

"Reinforcement learning", **STaR**, self-instruct, self-play, …

Training ← Search

Using Search to Improve Learning

"Write me python code to compute the fibonacci sequence" → LLM

```python
def fibonacci_3(n):
    a, b = 1, 1  # Incorrect initialization
    for _ in range(n):
        a = b
        b = a + b
    return a  # Returns an incorrect Fibonacci value
```

→ Verifier → incorrect

```python
def fibonacci_2(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci_2(n - 1) + fibonacci_2(n - 2)
```

→ Verifier → correct

Training ← Search

Using Search to Improve Learning

x **y**

"Write me python code to compute the fibonacci sequence" ,

```
def fibonacci_2(n):
    if n == 0:
        return 0
    elif n == 1:
        return 1
    else:
        return fibonacci_2(n - 1) + fibonacci_2(n - 2)
```

...

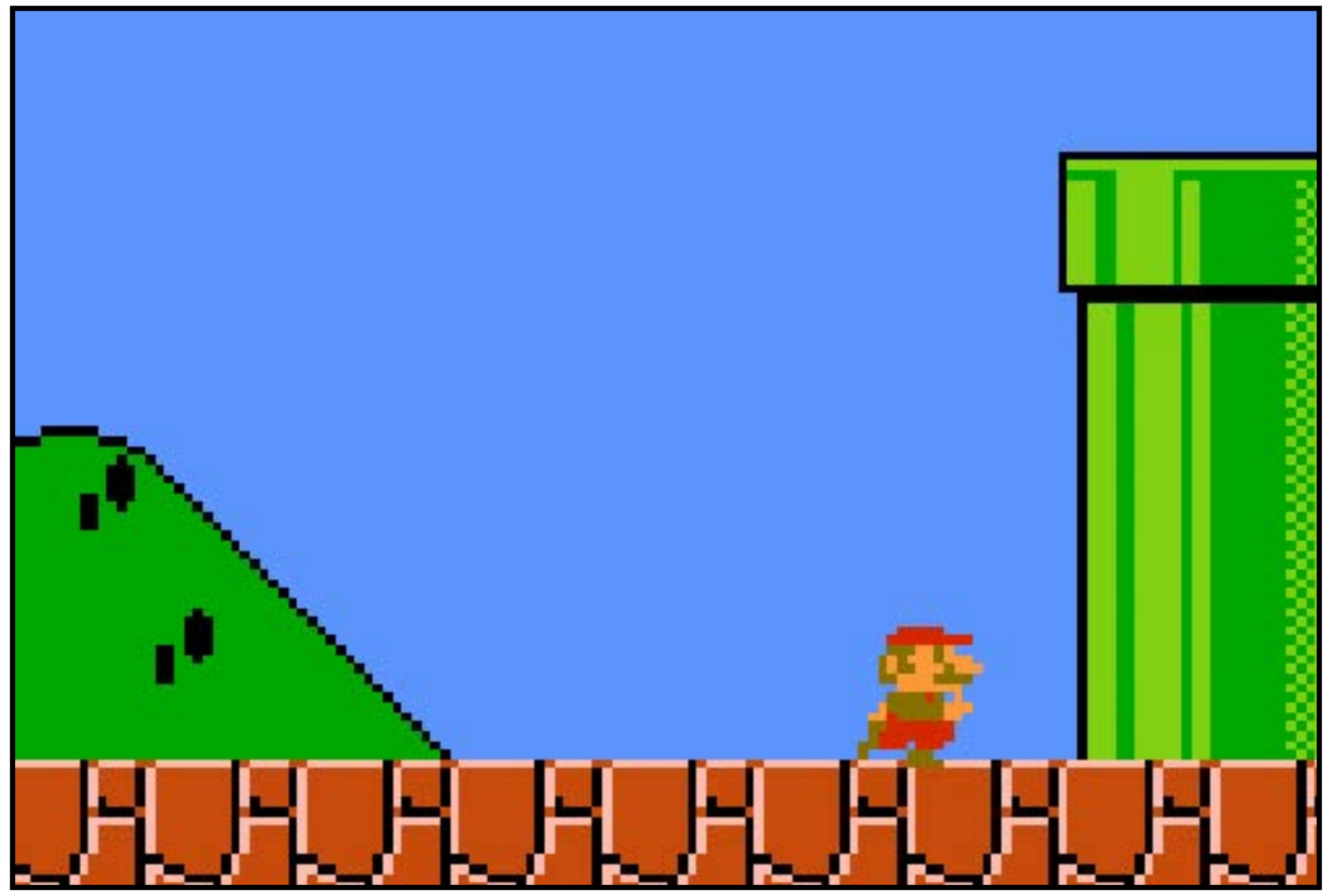
→ Supervised learning → LLM’

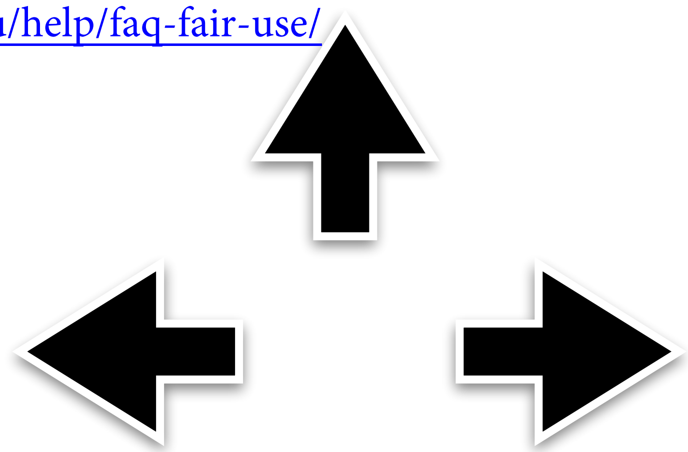
Training ⟷ Search

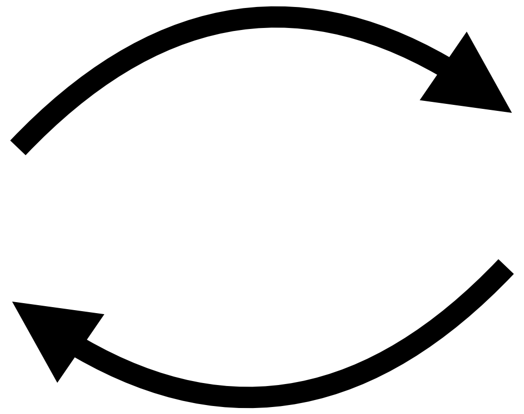
Using Search to Improve Learning and Using Learning to Improve Search

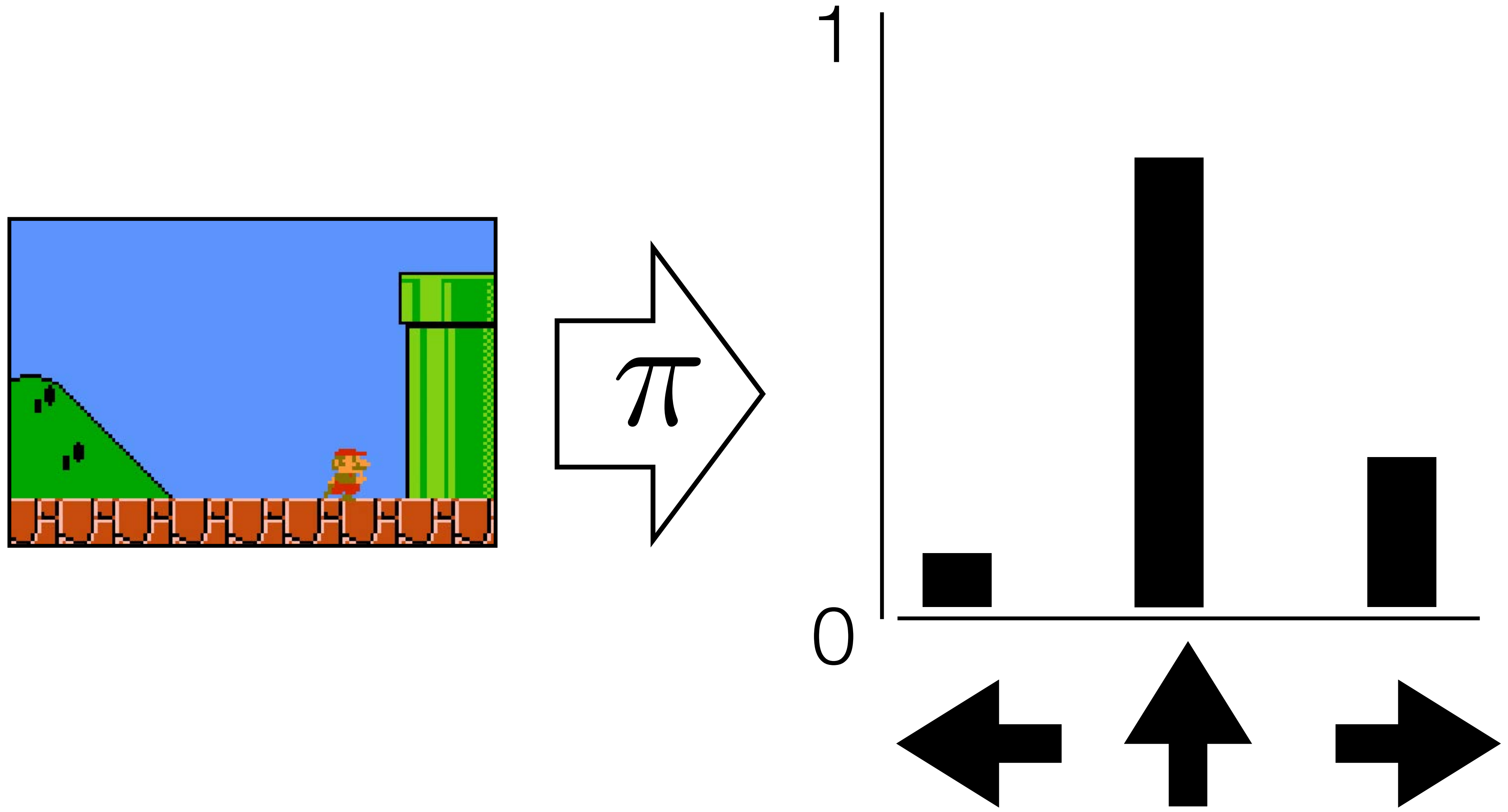
1. Search for random action sequences that lead to high rewards (winning).

2. Train a network (policy) that will tend to do those winning moves.





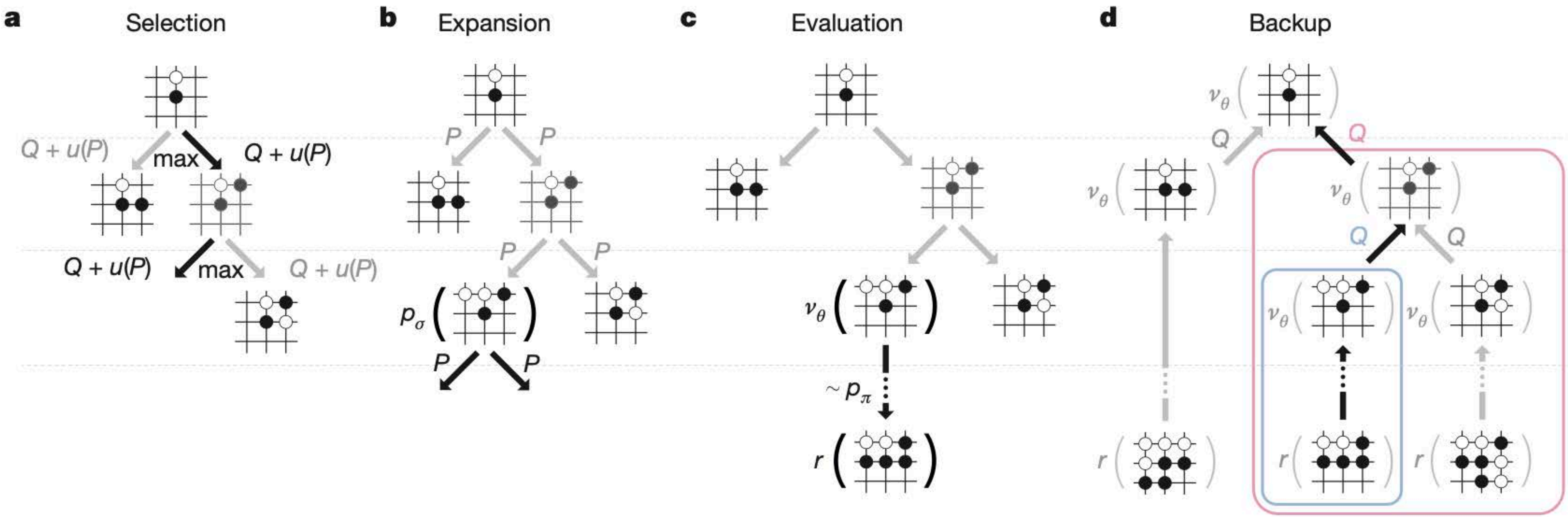




3. Repeat, guiding search using the policy to explore intelligently.

Training ⟷ Search

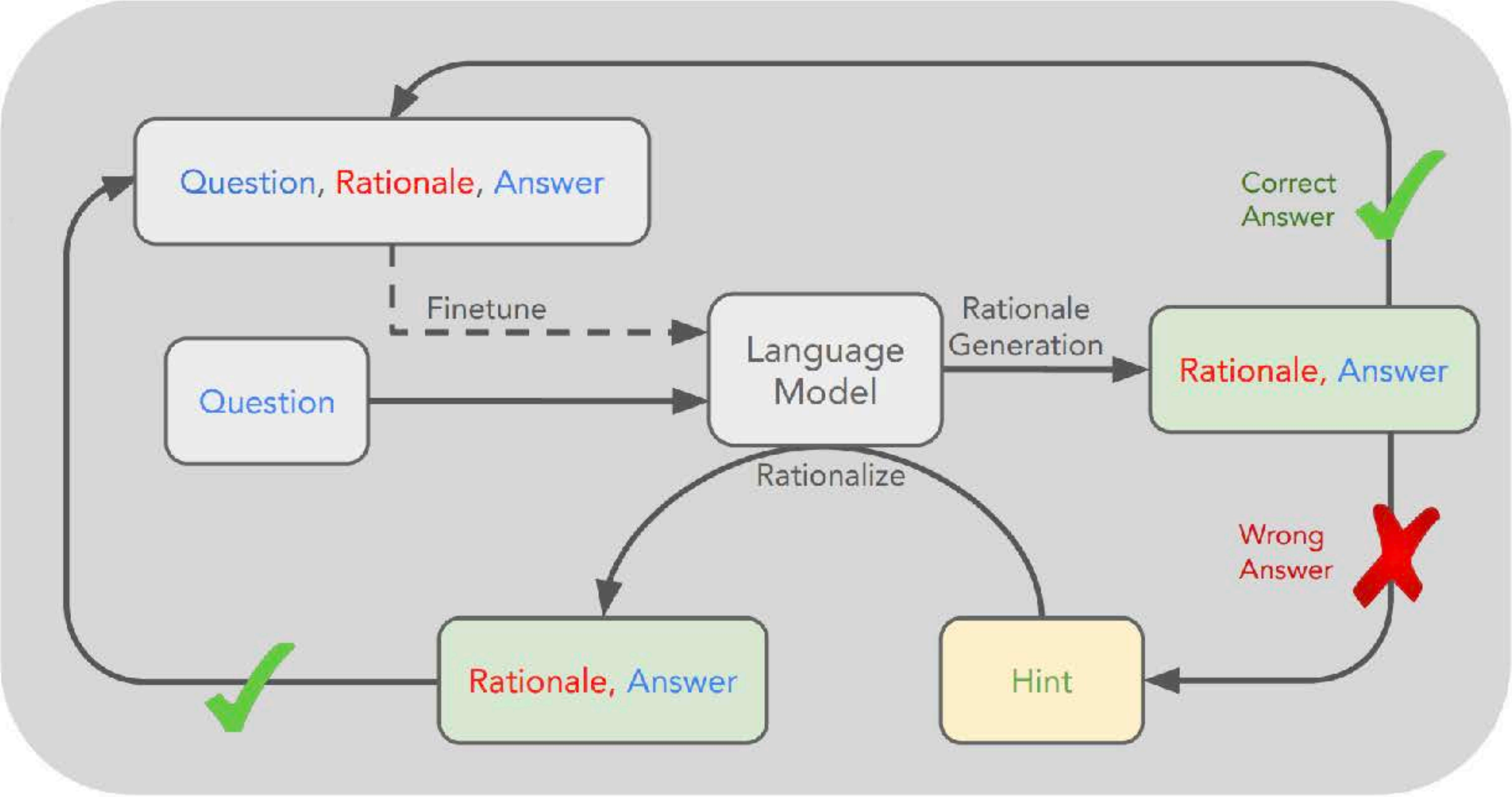
Using Search to Improve Learning and Using Learning to Improve Search



[AlphaGo: Silver*, Huang* et al, Nature 2016]

Training ⟷ Search

STaR



Q: What can be used to carry a small dog?
Answer Choices:
(a) swimming pool
(b) basket
(c) dog show
(d) backyard
(e) own home
A: The answer must be something that can be used to carry a small dog. Baskets are designed to hold things. Therefore, the answer is basket (b).

[Zelikman*, Wu*, Mu, Goodman, 2022]

Training ⟷ Search

o1?

1. Search for CoTs that solve the problem (randomly sample different chains)

2. Finetune the LLM to preferentially output CoTs that led to high reward

```
A: The answer must be
something that can be
used to carry a small
dog. Baskets are
designed to hold things.
Therefore, the answer
is basket (b).
```

...

3. Repeat, now the LLM outputs better CoTs

x **y**

```
Q: What can be used
to carry a small dog?
Answer Choices:
(a) swimming pool
(b) basket
(c) dog show
(d) backyard
(e) own home
```

,

```
A: The answer must be
something that can be
used to carry a small
dog. Baskets are
designed to hold things.
Therefore, the answer
is basket (b).
```

4. At deployment time, you can choose how much search to do on top of your fine-tuned LLM

Speculation from Sasha Rush: https://www.youtube.com/watch?v=6PEJ96k1kiw&ab_channel=SashaRush%F0%9F%A4%97