MACHINE LEARNING DEPARTMENT

Carnegie Mellon University

Neural Cognitive Architectures for Never-Ending Learning

Anthony Platanios

e.a.platanios@cs.cmu.edu

Thesis Committee:

Tom Mitchell (Chair)	[Carnegie Mellon University]
Graham Neubig	[Carnegie Mellon University]
Rich Caruana	[Microsoft Research]
Eric Horvitz	[Microsoft Research]

GOOGLE \ SCIENCE \ TECH

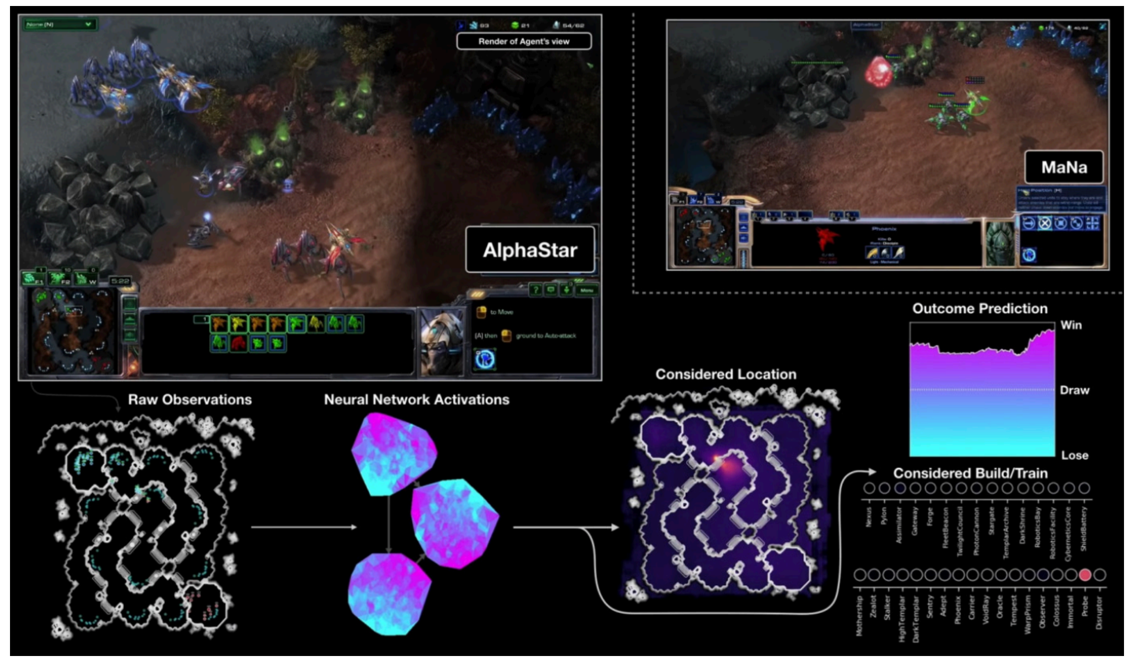
DeepMind’s Go-playing AI doesn’t need human help to beat us anymore

The company’s latest AlphaGo AI learned superhuman skills by playing itself over and over

By James Vincent | Oct 18, 2017, 1:00pm EDT

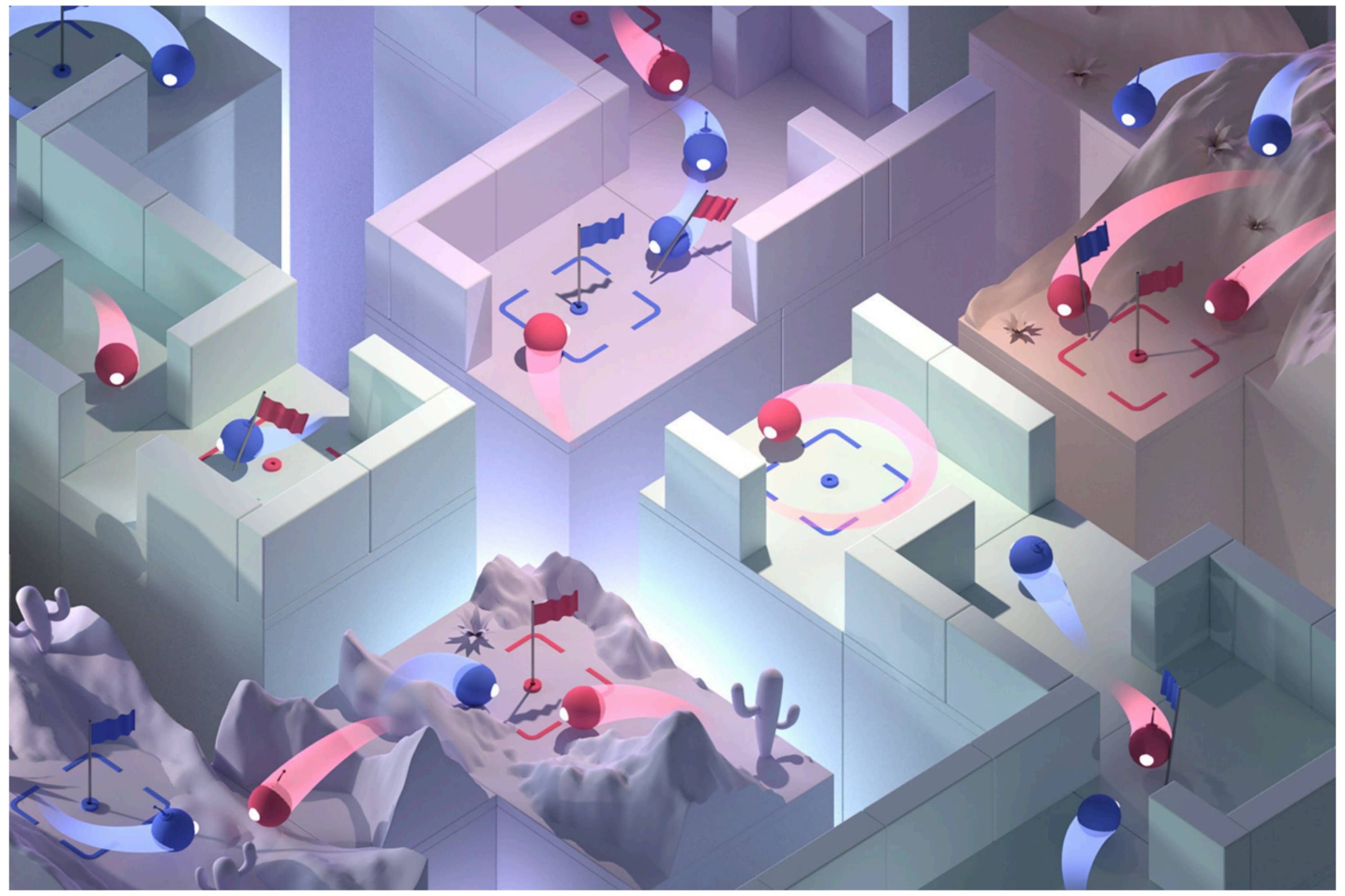
StarCraft II-playing AI AlphaStar takes out pros undefeated

Devin Coldewey @techcrunch / 5 months ago Comment



DeepMind Can Now Beat Us at Multiplayer Games, Too

Chess and Go were child’s play. Now A.I. is winning at capture the flag. Will such skills translate to the real world?



DeepMind

By Cade Metz

May 30, 2019

56

When Is Technology Too Dangerous to Release to the Public?

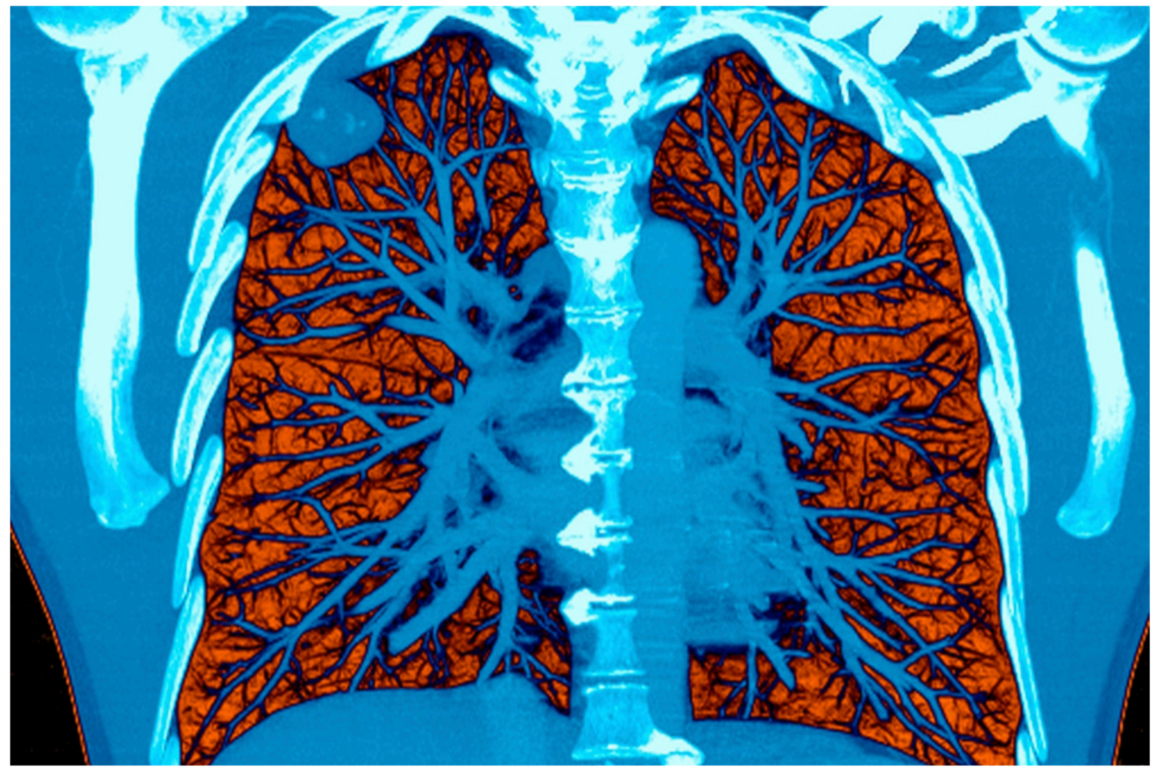
A new text-generating algorithm has reignited a long-running debate.

By AARON MAK

FEB 22, 2019 • 5:56 PM

A.I. Took a Test to Detect Lung Cancer. It Got an A.

Artificial intelligence may help doctors make more accurate readings of CT scans used to screen for lung cancer.



A colored CT scan showing a tumor in the lung. Artificial intelligence was just as good, and sometimes better, than doctors in diagnosing lung tumors in CT scans, a new study indicates. Voisin/Science Source

By Denise Grady

May 20, 2019

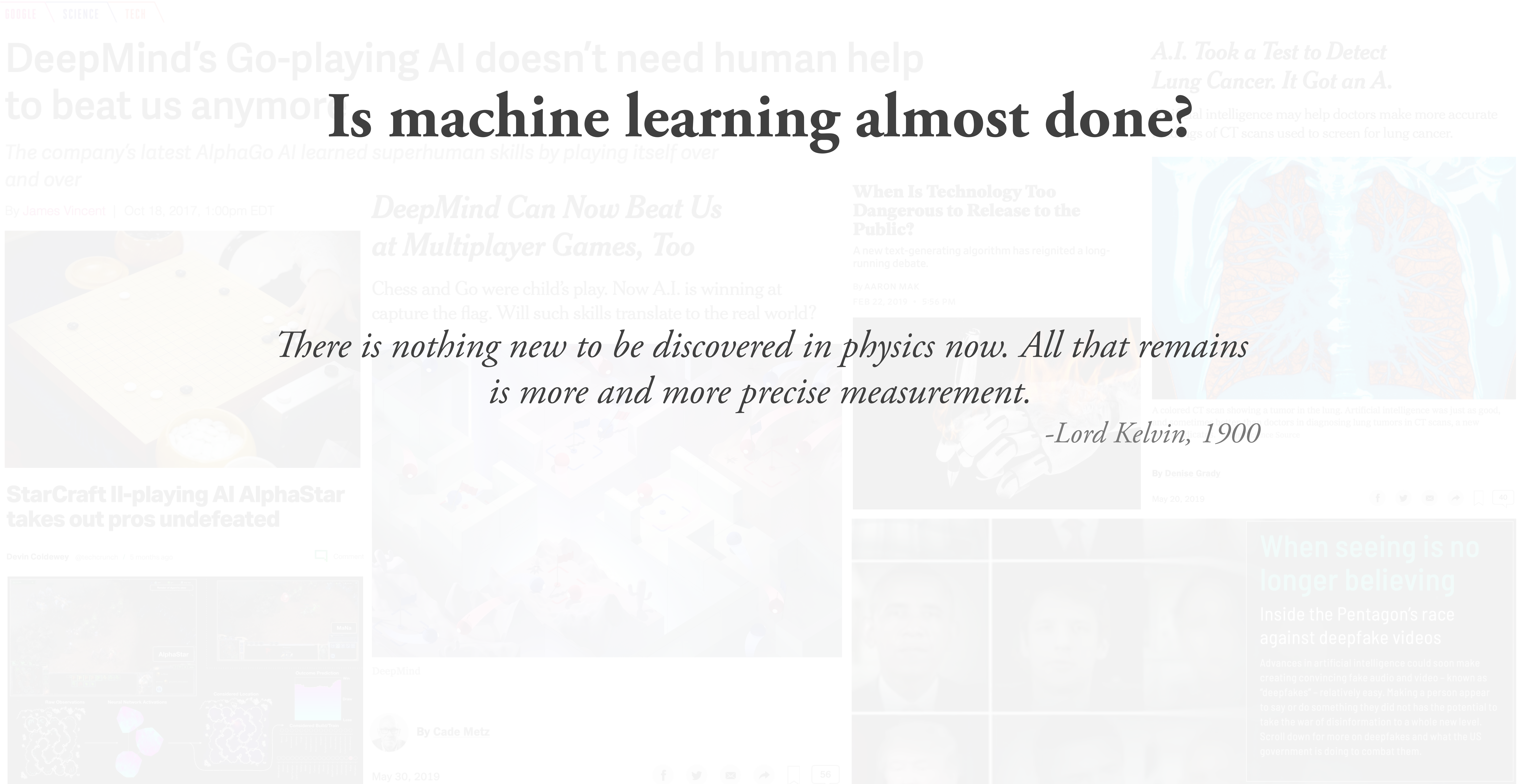
40

When seeing is no longer believing

Inside the Pentagon’s race against deepfake videos

Advances in artificial intelligence could soon make creating convincing fake audio and video – known as “deepfakes” – relatively easy. Making a person appear to say or do something they did not has the potential to take the war of disinformation to a whole new level. Scroll down for more on deepfakes and what the US government is doing to combat them.

Is machine learning almost done?



There is nothing new to be discovered in physics now. All that remains is more and more precise measurement.

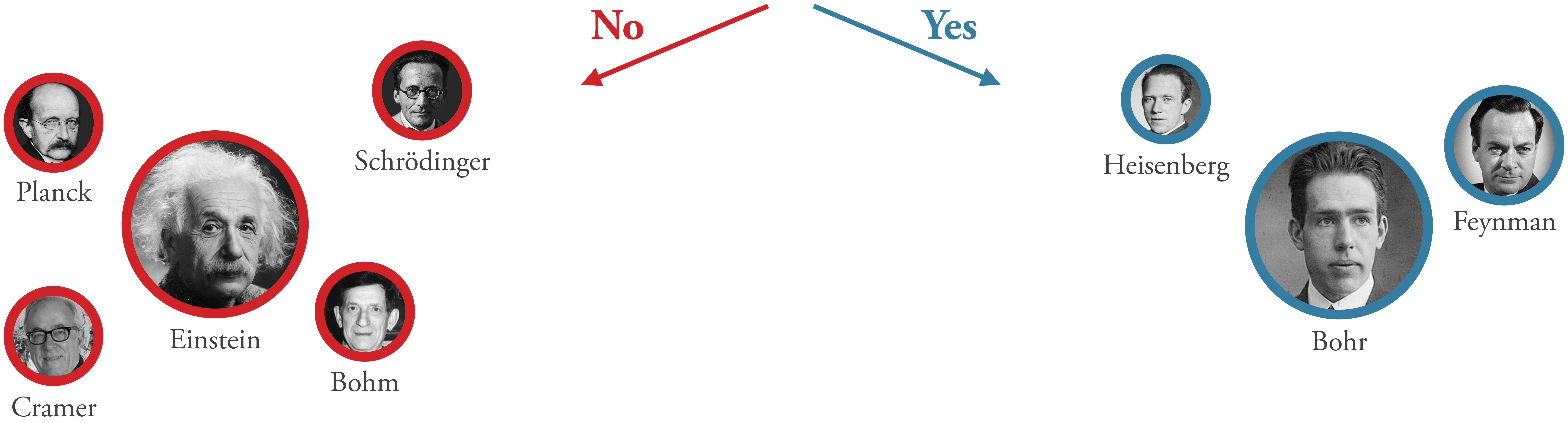
-Lord Kelvin, 1900

What is wrong?

Limited ability for *general learning and intelligence.*

1 Cannot handle *supervision noise and ambiguity.*

Is quantum physics probabilistic?



What is wrong?

Limited ability for *general learning and intelligence.*

1. Cannot handle *supervision noise and ambiguity*.

2. Limited ability for *multi-task learning:*
   - Overfitting to a single task
   - Easy tasks are hurt, while hard tasks benefit
   - “Fast-learning” tasks are hurt, while “slow-learning” tasks benefit
   - Cannot handle a dynamically varying number of tasks

3. Limited ability for *unsupervised learning*.

Thesis Statement

> “*A computer system with an architecture inspired by human cognition can learn to continuously solve multiple problems that can grow in number over time, across multiple distinct perception and action modalities, and from multiple noisy sources of supervision combined with self-supervision. Furthermore, its experience from learning to solve past problems can be leveraged to learn to solve future ones.*”

Goals

Formalize never-ending learning and the notion of a neural cognitive architecture.

Design a neural cognitive architecture that is inspired by human cognition.

Evaluate the capabilities of the proposed architecture.

Limited ability for *general learning and intelligence*.

1. Cannot handle *supervision noise and ambiguity*.
2. Limited ability for *multi-task learning*.
3. Limited ability for *unsupervised learning*.

What is wrong?

Goals

Formalize never-ending learning and the notion of a neural cognitive architecture.

Design a neural cognitive architecture that is inspired by human cognition.

Evaluate the capabilities of the proposed architecture.

1. Cannot handle *supervision noise and ambiguity*.
2. Limited ability for *multi-task learning*.
3. Limited ability for *unsupervised learning*.
4. Limited ability for *general learning and intelligence*.

What is wrong?

Goals

Formalize never-ending learning and the notion of a neural cognitive architecture.

Design a neural cognitive architecture that is inspired by human cognition.

Evaluate the capabilities of the proposed architecture.

1. Learning from Noisy Labels
   supervision noise and ambiguity
2. Contextual Parameter Generation
   multi-task learning
3. Self-Reflection
   unsupervised learning
4. Unified Architecture
   general learning and intelligence

} Contributions

Outline

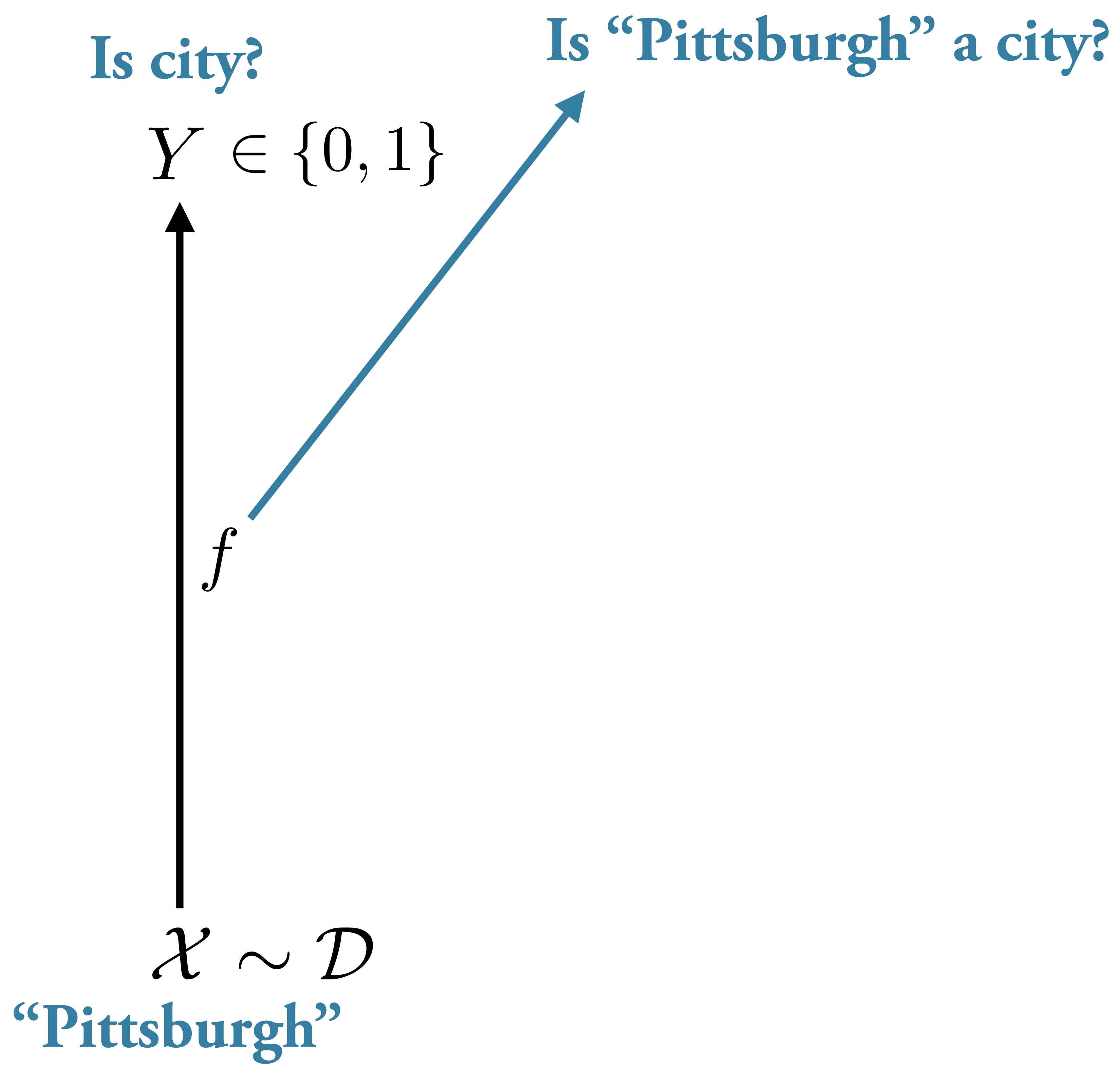
Part 1: Proposed Approach

1. Learning from Noisy Labels
   supervision noise and ambiguity
2. Contextual Parameter Generation
   multi-task learning
3. Self-Reflection
   unsupervised learning
4. Unified Architecture
   general learning and intelligence

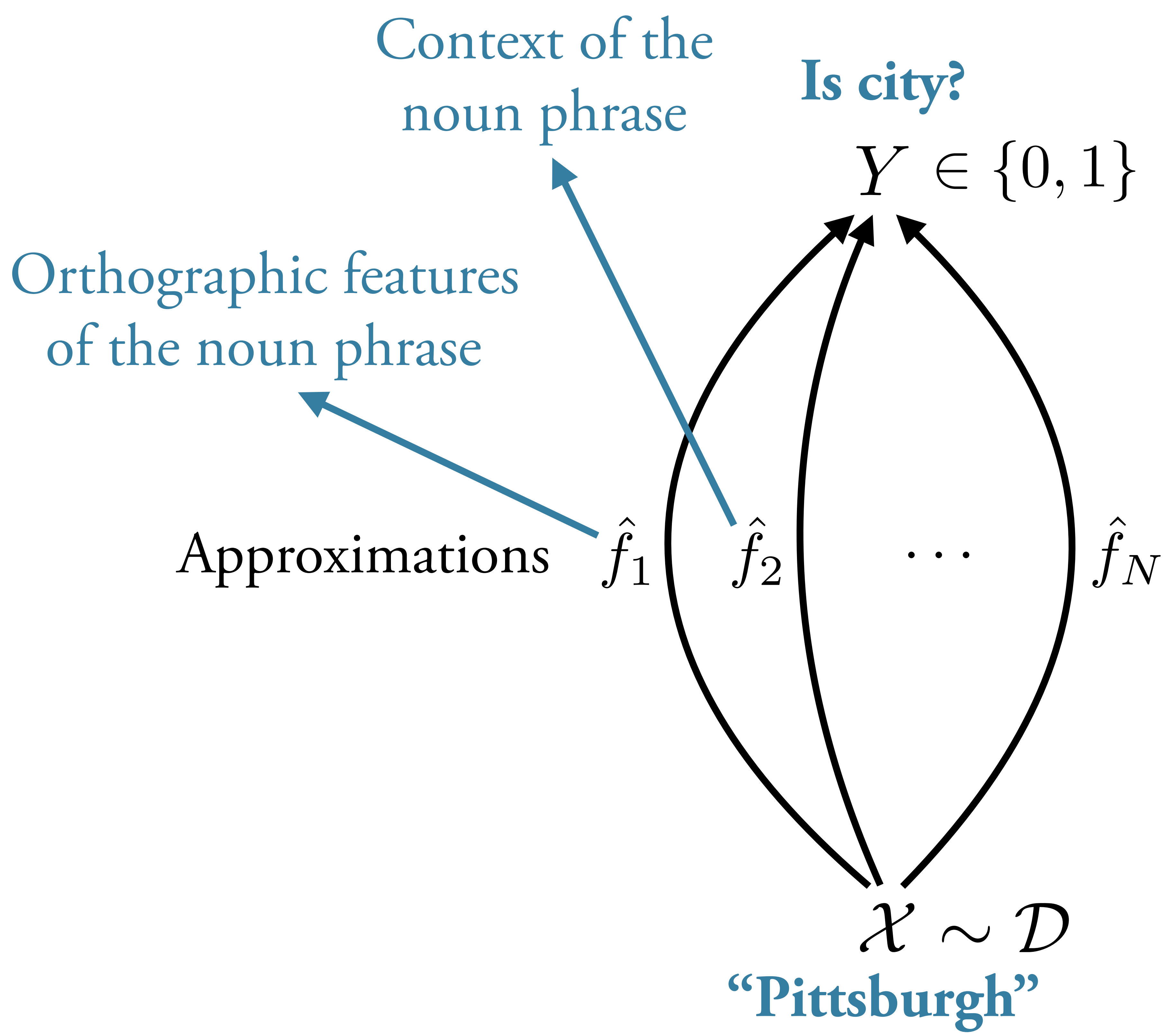
Part 2: Evaluation

Part 3: Timeline

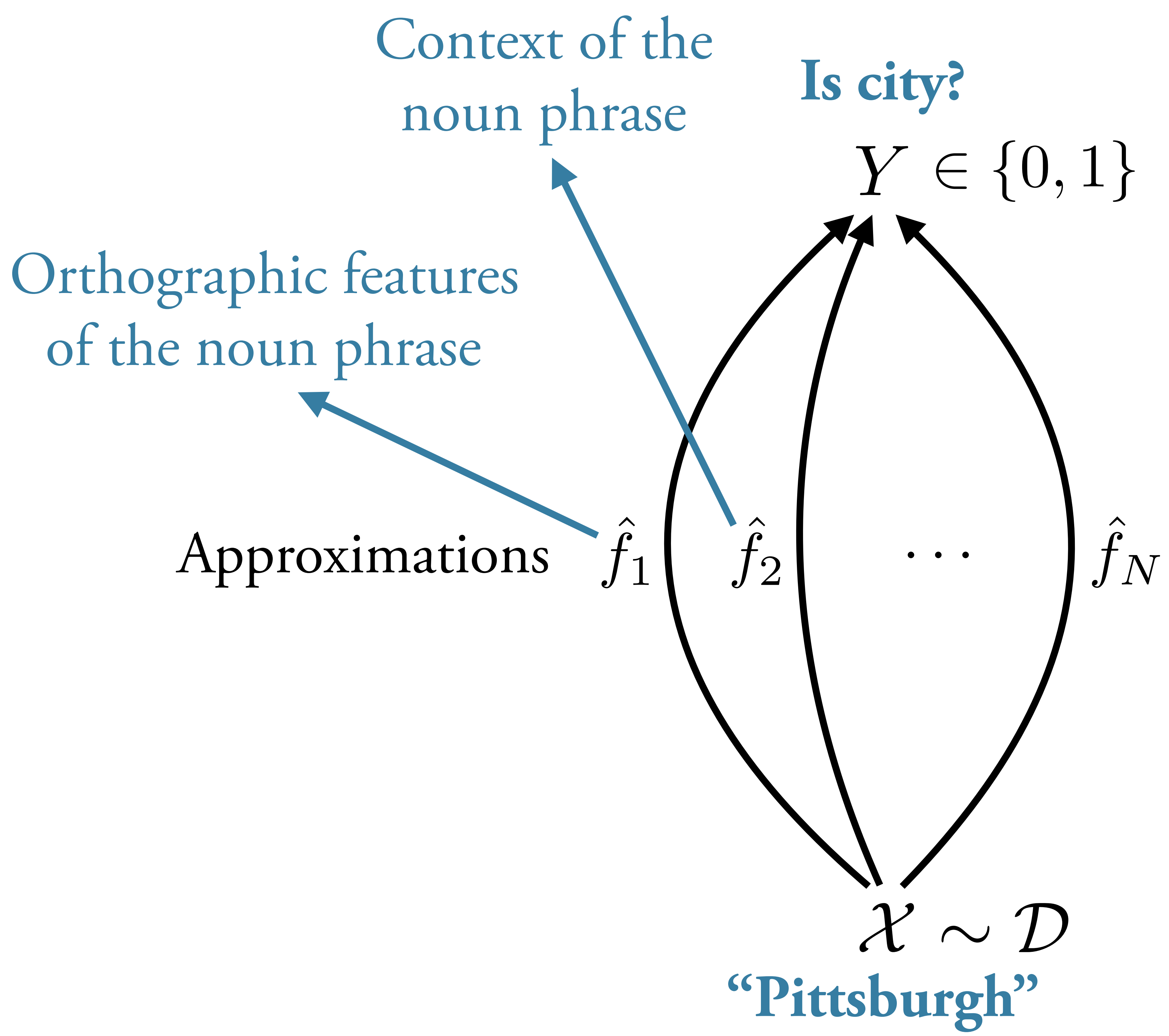
Learning from Noisy Labels



Learning from Noisy Labels



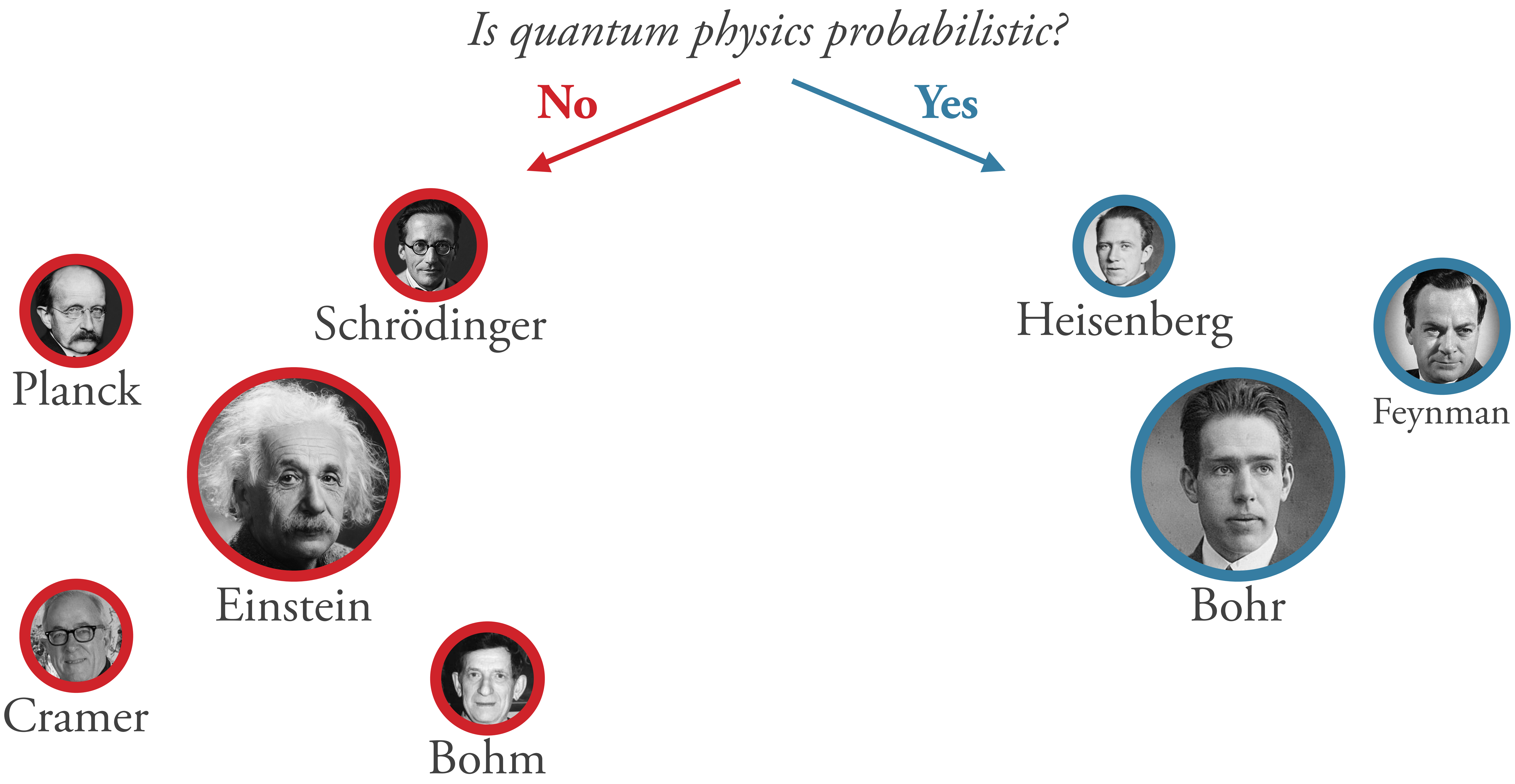
Learning from Noisy Labels



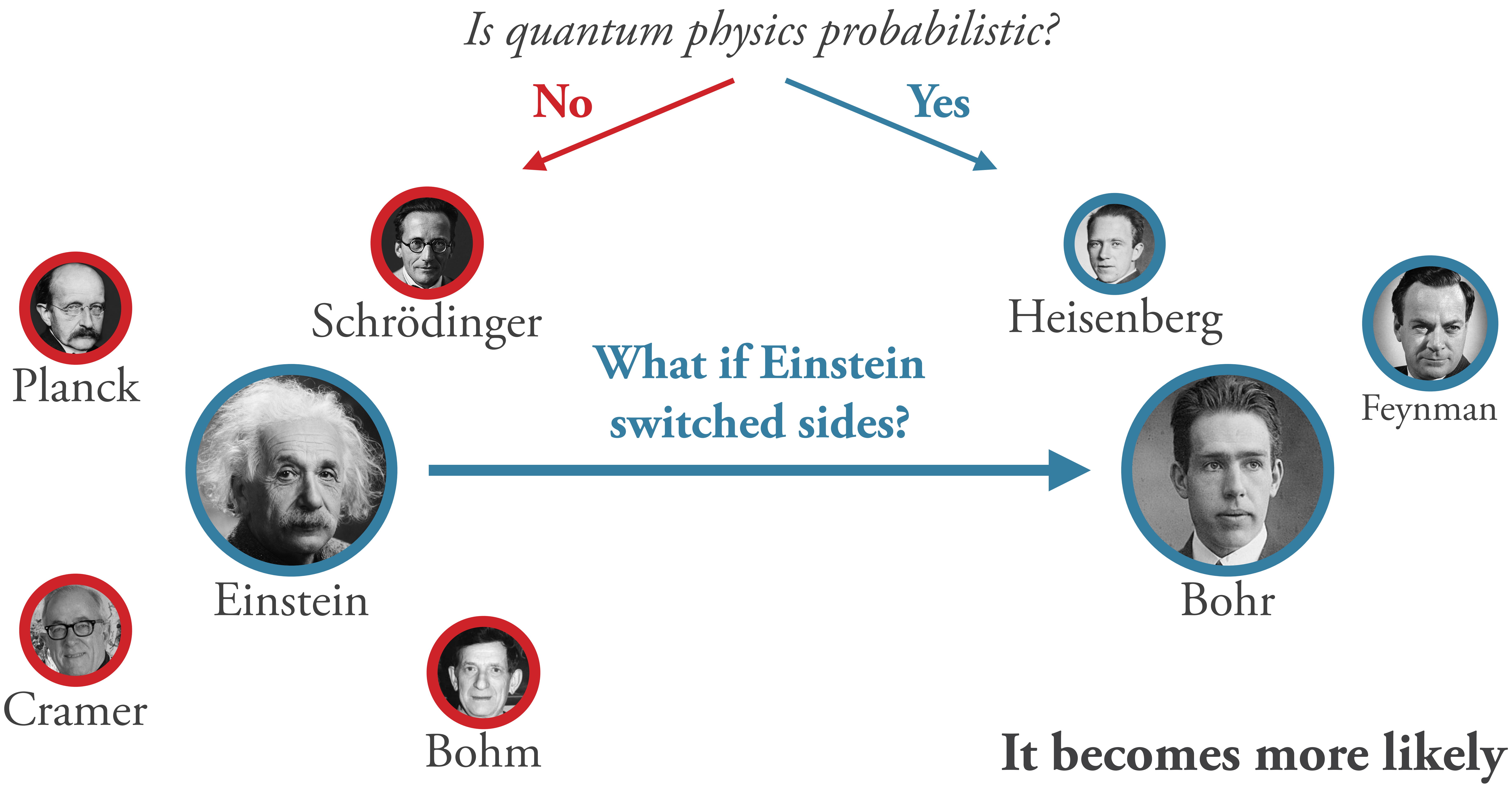
What would a human do?

Learning from Noisy Labels

What would a human do?



Learning from Noisy Labels



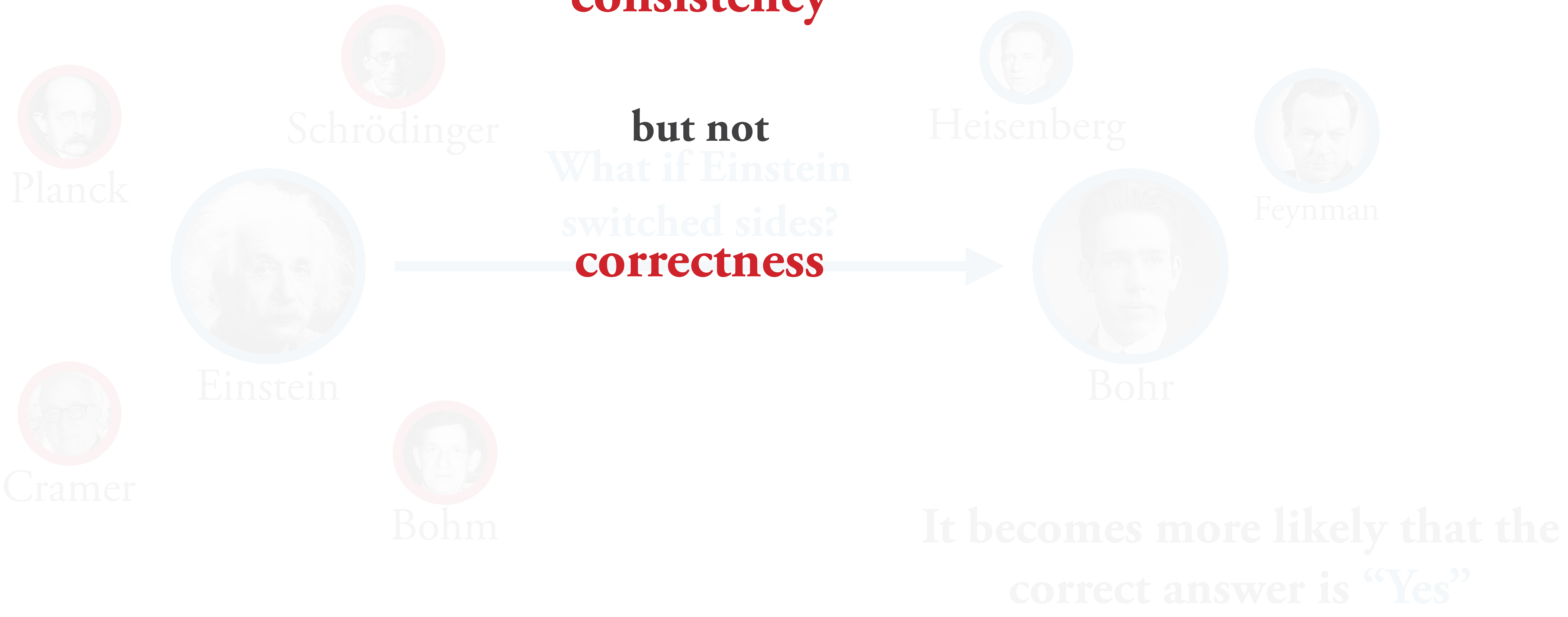
It becomes more likely that the correct answer is "Yes"

Using **only unlabeled data** we can measure

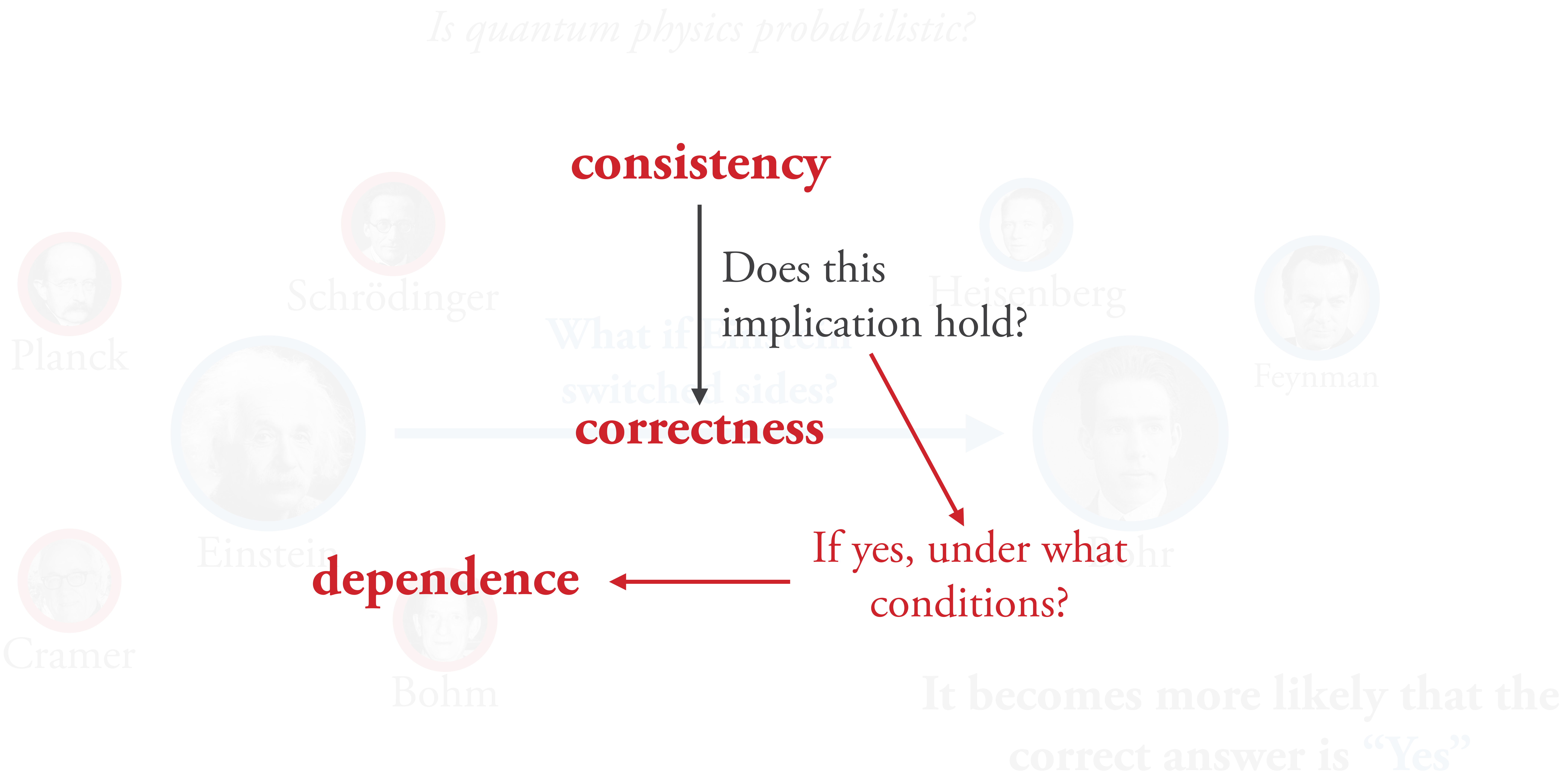
consistency

but not

correctness



Learning from Noisy Labels



Learning from Noisy Labels

We developed a few methods based on this intuition:

1. Agreement Rates: Optimization problem with linear constraints.
2. Bayesian Models: Generative model that accounts for our observations.

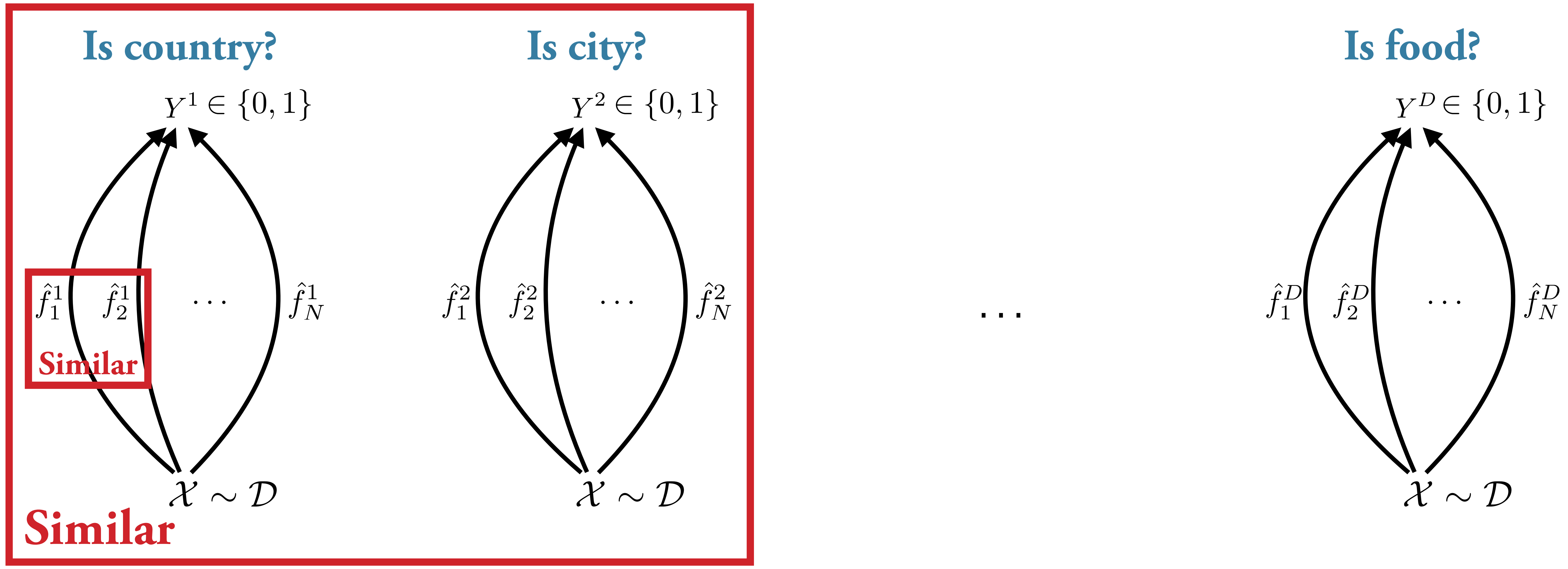
[**Platanios**, Dubey, Mitchell, ICML 2016]

Learning from Noisy Labels

We developed a few methods based on this intuition:

1. <u>Agreement Rates:</u> Optimization problem with linear constraints.
2. <u>Bayesian Models:</u> Generative model that accounts for our observations.

What if we have multiple questions?



We can share information using a clustering prior.

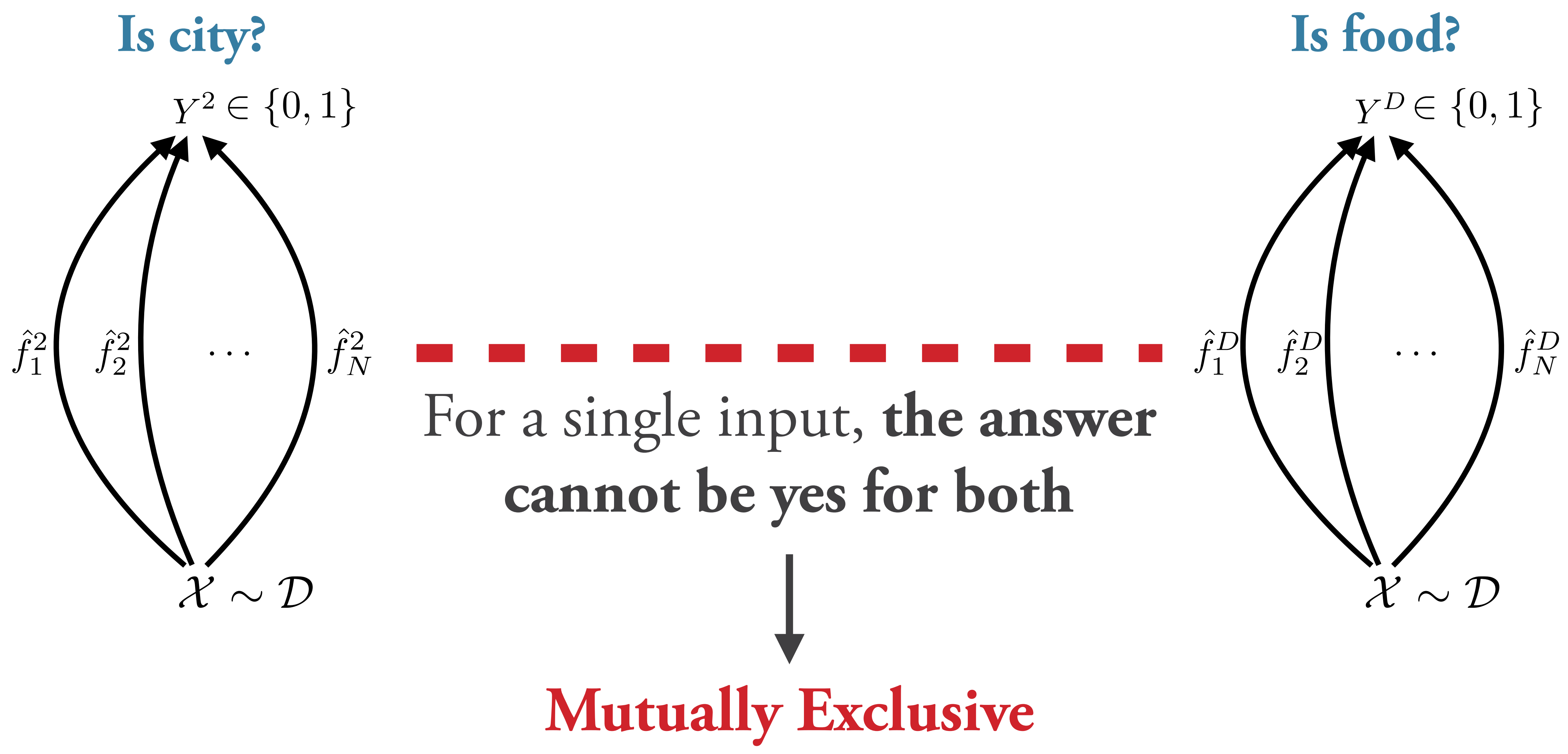
[**Platanios**, Dubey, Mitchell, ICML 2016]

Learning from Noisy Labels

We developed a few methods based on this intuition:

1. Agreement Rates: Optimization problem with linear constraints.
2. Bayesian Models: Generative model that accounts for our observations.

What if we know of some logical constraints?



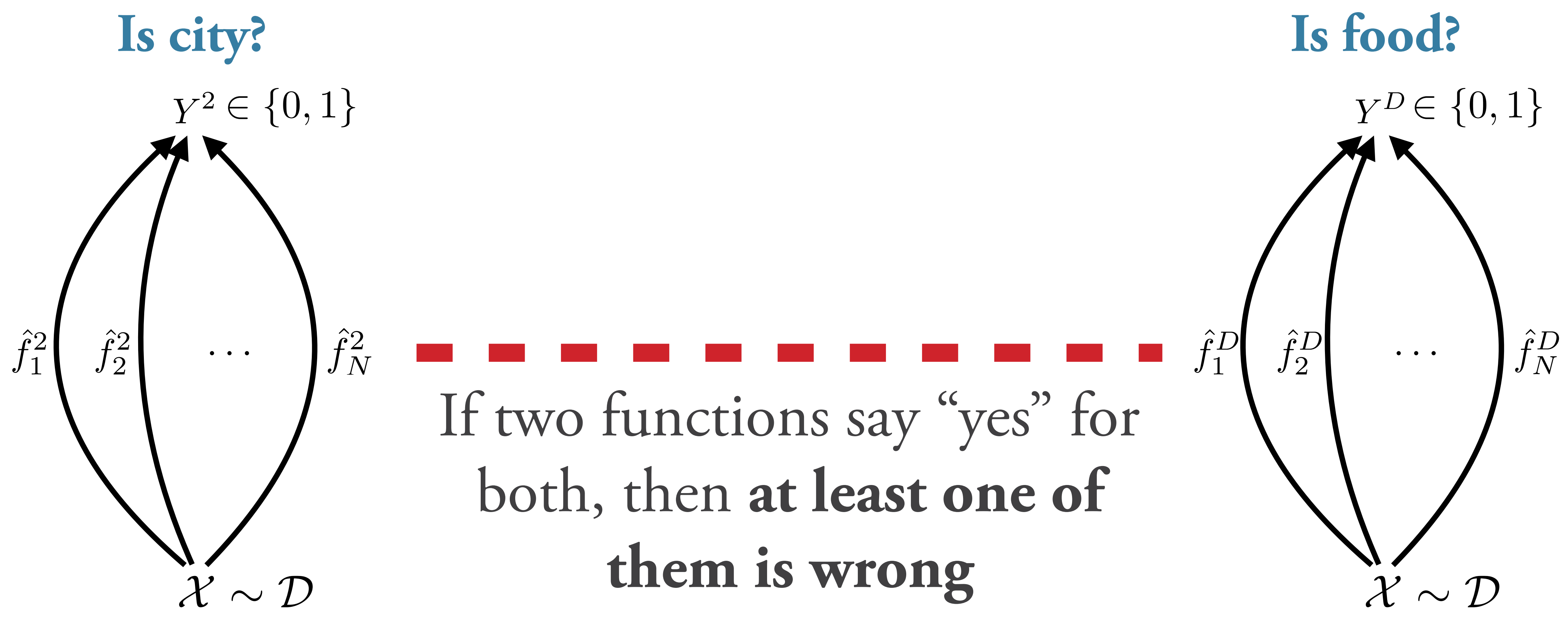
[**Platanios**, Dubey, Mitchell, ICML 2016]

Learning from Noisy Labels

We developed a few methods based on this intuition:

1. Agreement Rates: Optimization problem with linear constraints.
2. Bayesian Models: Generative model that accounts for our observations.

What if we know of some logical constraints?

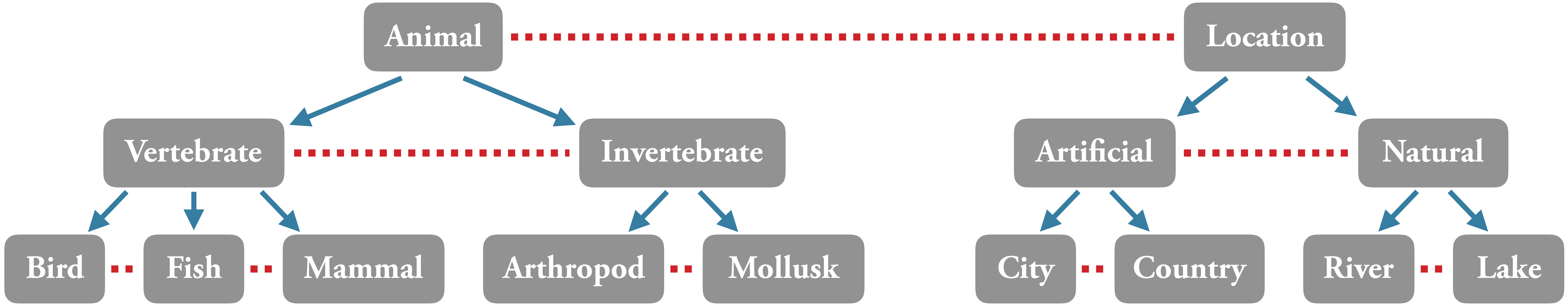


[**Platanios**, Dubey, Mitchell, ICML 2016]

Learning from Noisy Labels

We developed a few methods based on this intuition:

1. Agreement Rates: Optimization problem with linear constraints.
2. Bayesian Models: Generative model that accounts for our observations.
3. Logic-Based Approach: Perform probabilistic logic inference in the presence of logical constraints.



[**Platanios**, Poon, Horvitz, Mitchell, NeurIPS 2017]

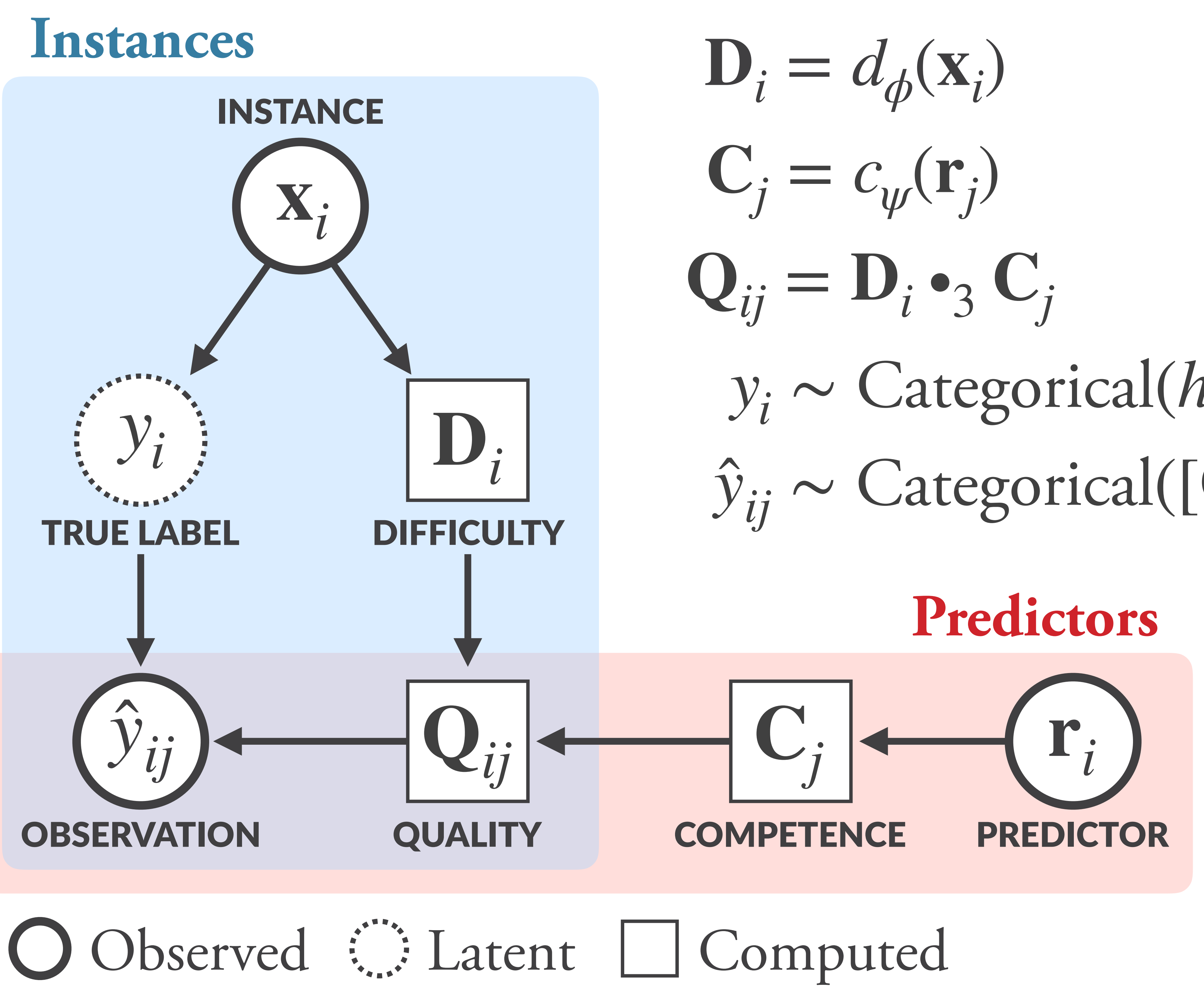
Learning from Noisy Labels

Can we merge the accuracy estimation methods with the model learning phase?

We learn:

- A model of the *ground truth*.
- A model of the items' *difficulties*.
- A model of the predictors' *competences*.

We derive an *Expectation-Maximization algorithm* to perform learning.



$$\mathbf{D}_i = d_\phi(\mathbf{x}_i)$$

$$\mathbf{C}_j = c_\psi(\mathbf{r}_j)$$

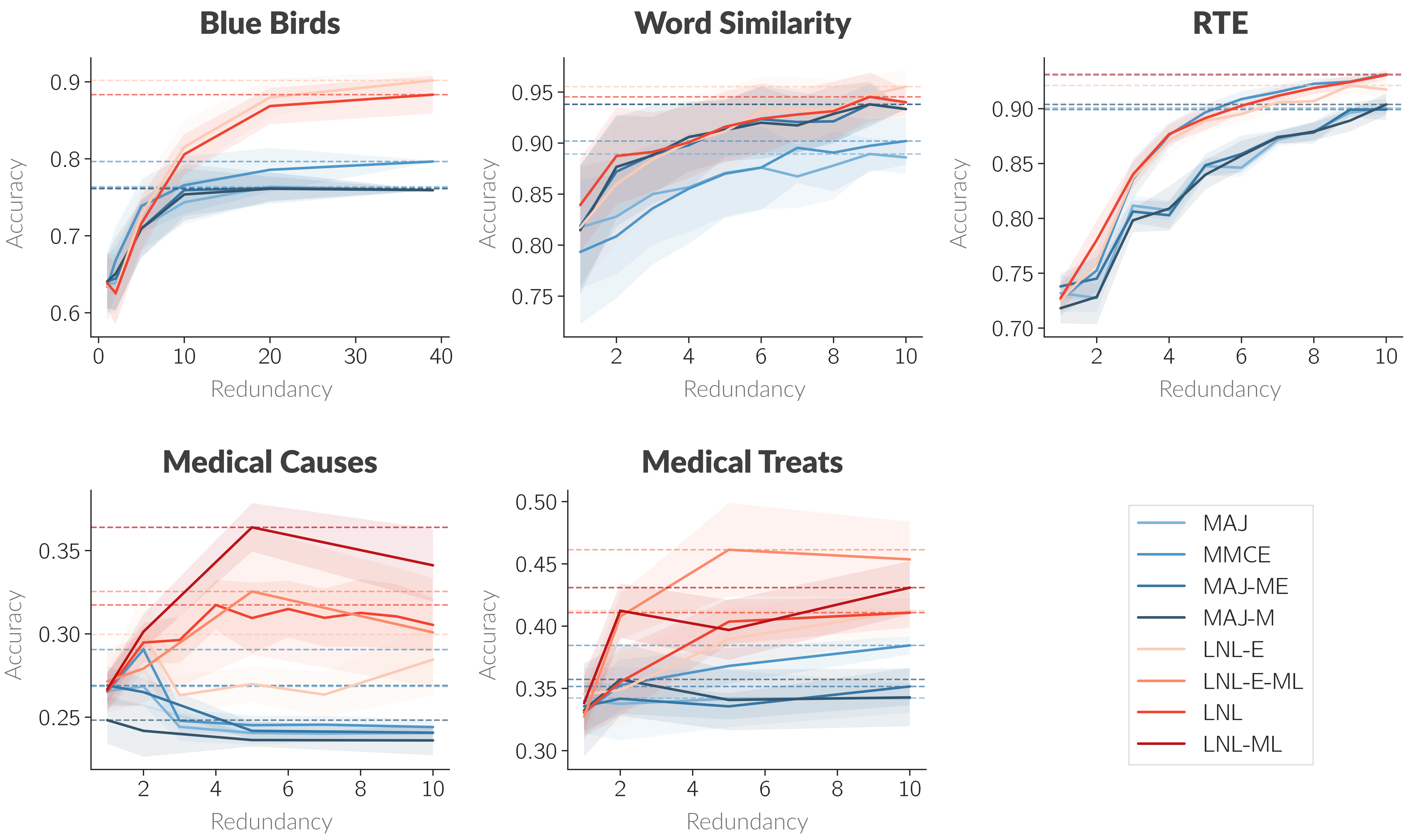
$$\mathbf{Q}_{ij} = \mathbf{D}_i \bullet_3 \mathbf{C}_j$$

$$y_i \sim \text{Categorical}(h_\theta(\mathbf{x}_i))$$

$$\hat{y}_{ij} \sim \text{Categorical}([\mathbf{Q}_{ij}]_{y_i -})$$

[**Platanios**, Al-Shedivat, Xing, Mitchell, under review for NeurIPS 2019]

Learning from Noisy Labels



[**Platanios**, Al-Shedivat, Xing, Mitchell, under review for NeurIPS 2019]

Outline

Part 1: Proposed Approach

1. Learning from Noisy Labels
   supervision noise and ambiguity
2. Contextual Parameter Generation
   multi-task learning
3. Self-Reflection
   unsupervised learning
4. Unified Architecture
   general learning and intelligence

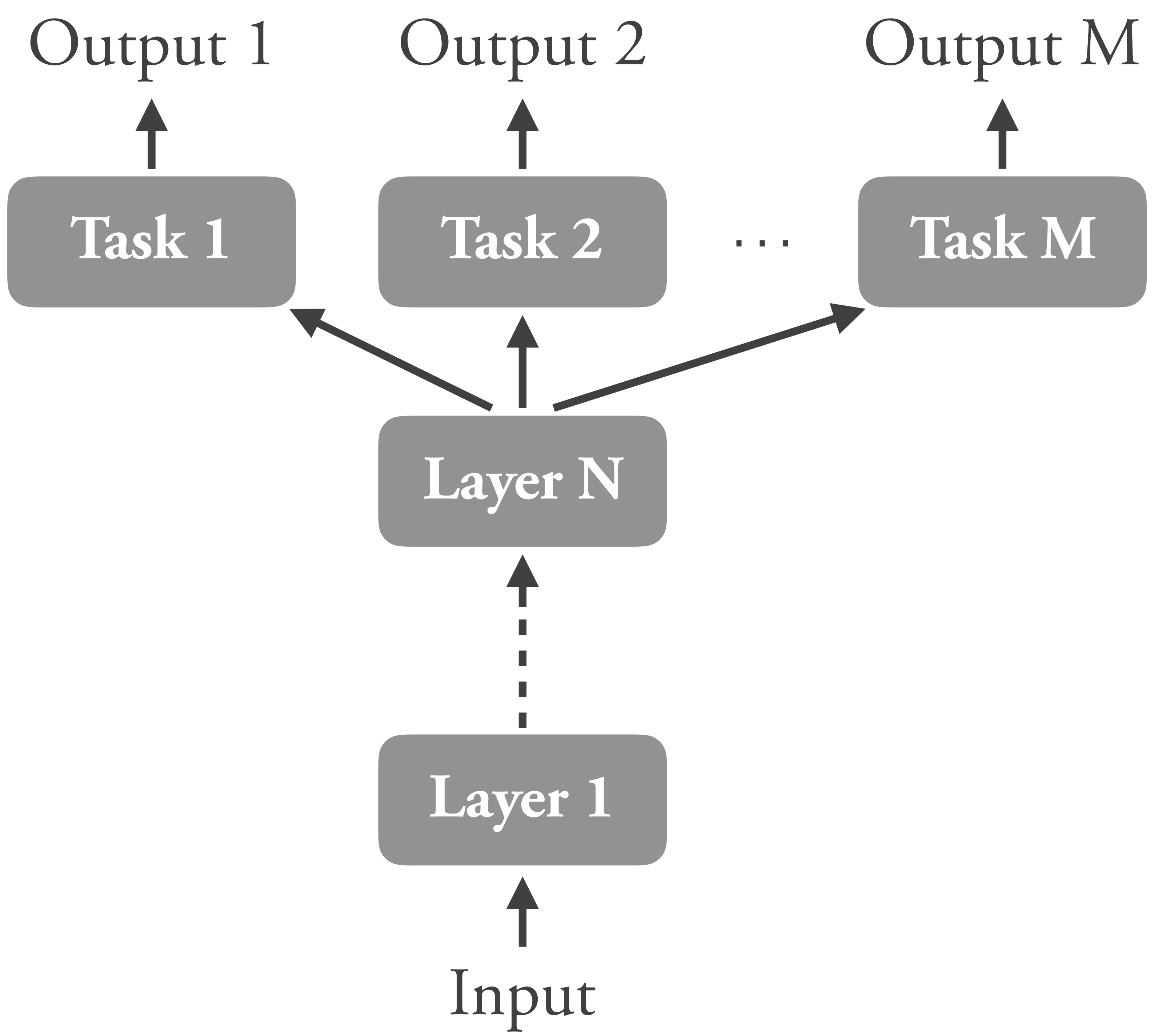
Part 2: Evaluation

Part 3: Timeline

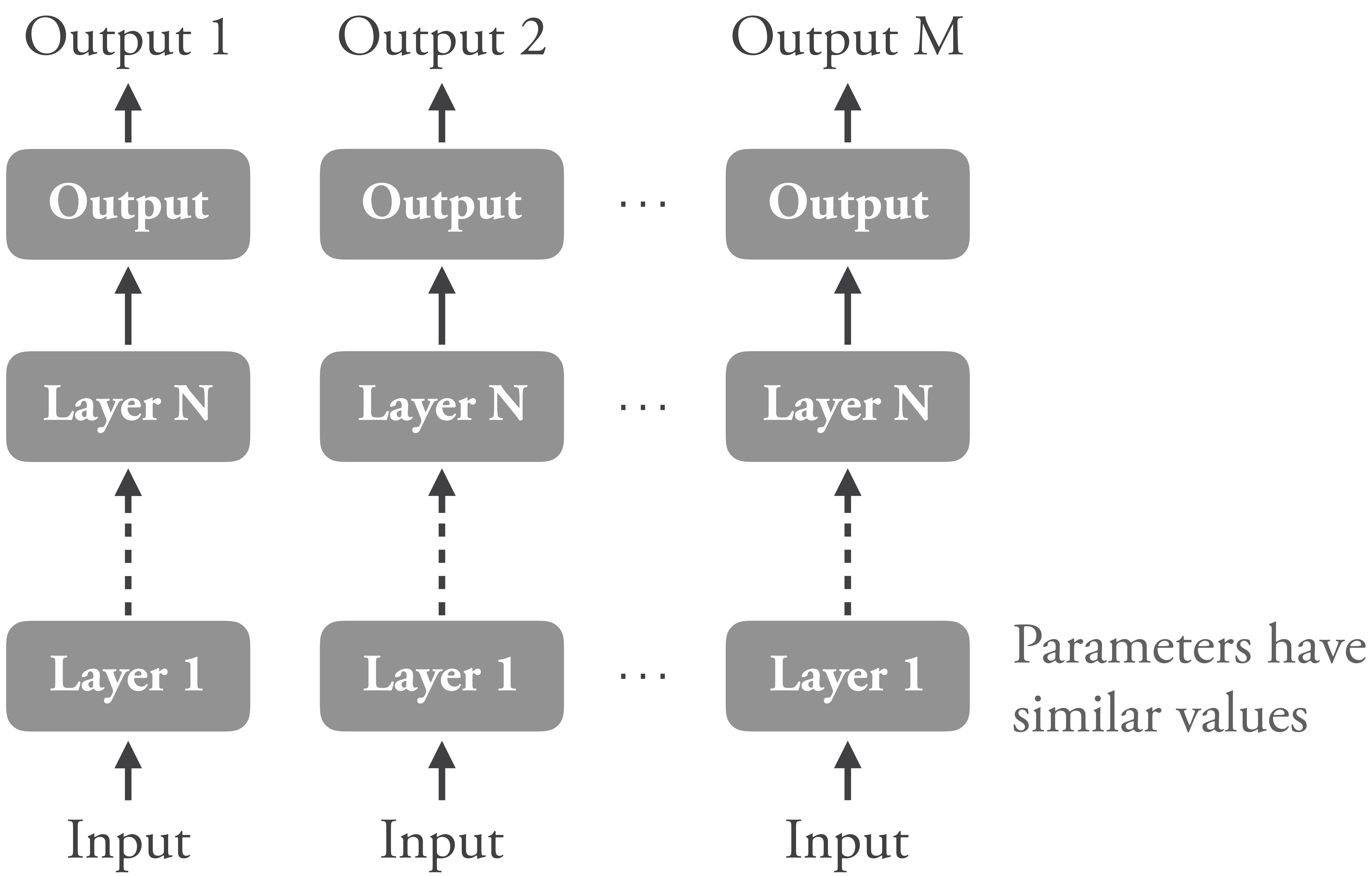
Multi-Task Learning

Multi-task learning is currently performed in one of two ways:

Hard Parameter Sharing



Soft Parameter Sharing



[Caruana, Machine Learning 1997 & Ruder, CoRR 2017]

Multi-task learning is currently performed in one of two ways:

Hard Parameter Sharing

- Adding new tasks requires *re-training*.
- Hard to encode structure and share information across tasks.
- Highly prone to *negative transfer*.

Soft Parameter Sharing

- Adding new tasks requires *copying/re-training*.
- Hard to have tasks interact in a constructive manner — tasks typically only interact through *pre-training*.

[Caruana, Machine Learning 1997 & Ruder, CoRR 2017]

Contextual Parameter Generation

Multi-task learning is currently performed in one of two ways:

Hard Parameter Sharing

Soft Parameter Sharing

A never-ending learning system must support learning tasks that keep changing.

- Adding new tasks requires *re-training*.
- Hard to encode structure and share information across tasks.
- Highly prone to *negative transfer*.

- Adding new tasks requires *copying/re-training*.
- Hard to have tasks interact in a constructive manner — tasks typically only interact through *pre-training*.

Do not allow for task dependencies at inference time!
(e.g., task composition)

What do humans do?

[Caruana, Machine Learning 1997 & Ruder, CoRR 2017]

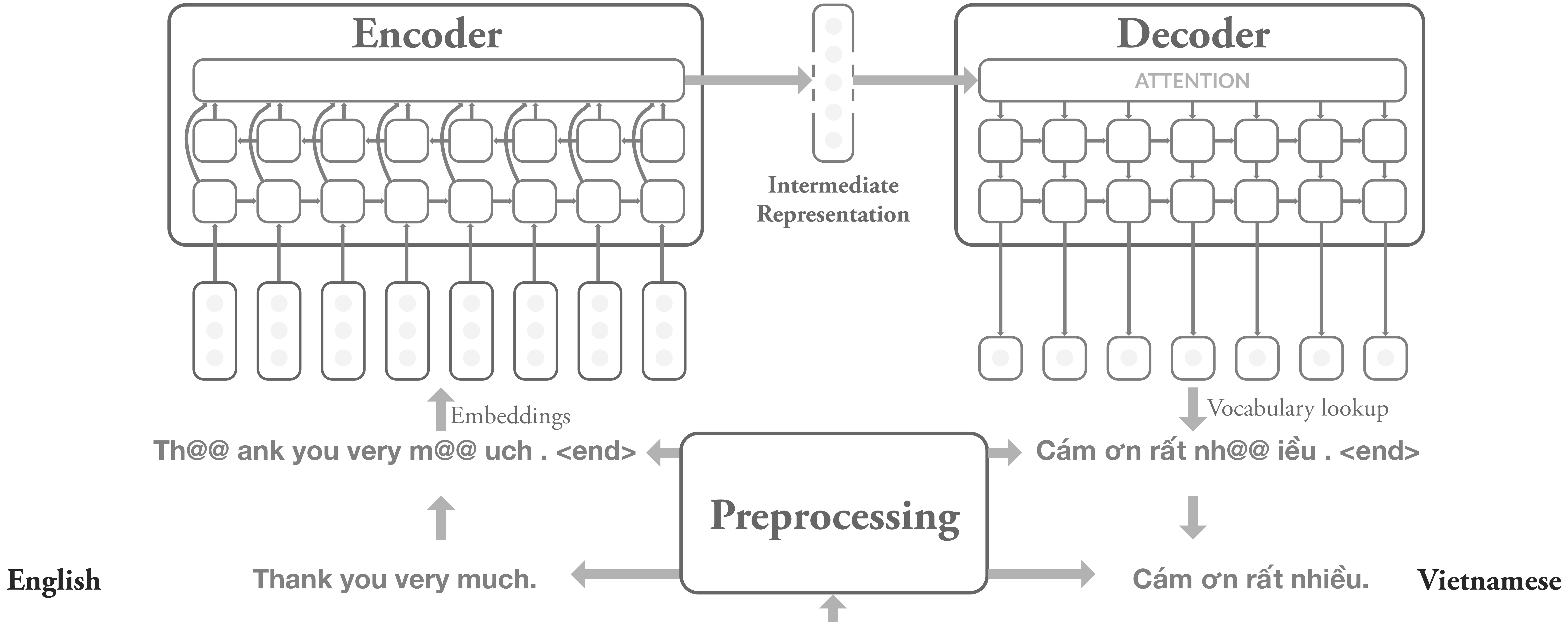
Contextual Parameter Generation

What if we learn *task representations* and feed them as *inputs*?

There have been some attempts. For example, in **machine translation (MT)**.

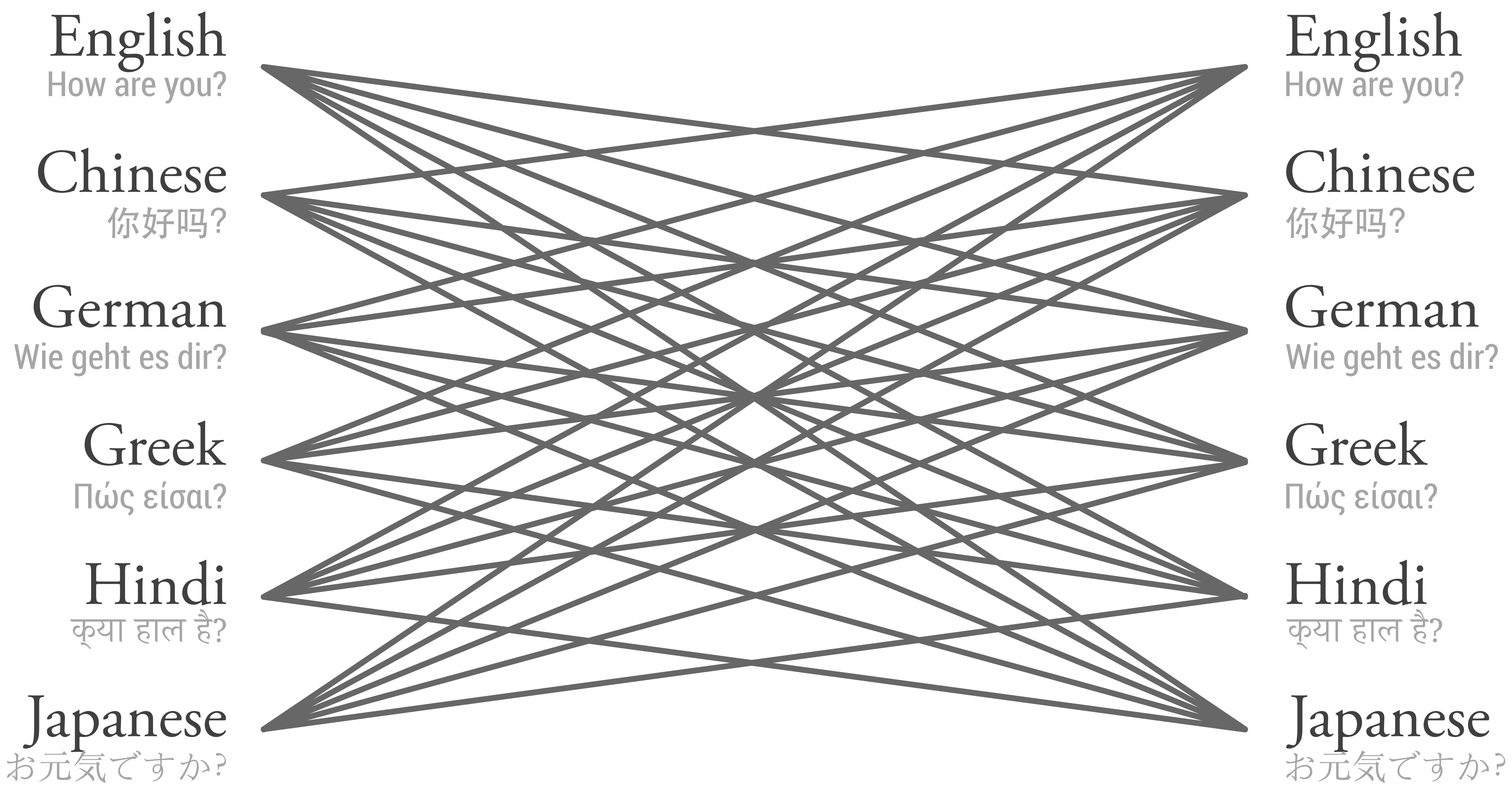
Contextual Parameter Generation

An MT system can translate from one language to another automatically, without human input.



Contextual Parameter Generation

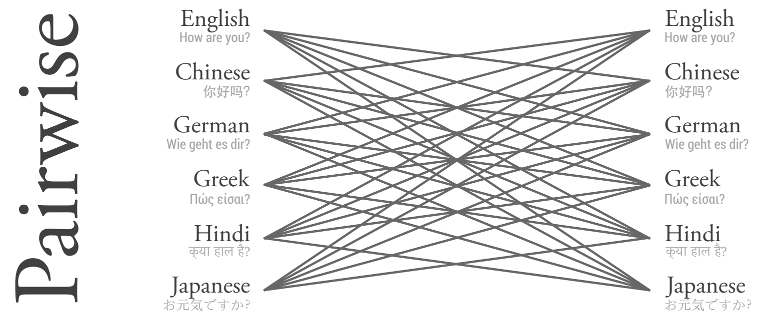
What about *multilingual translation*?

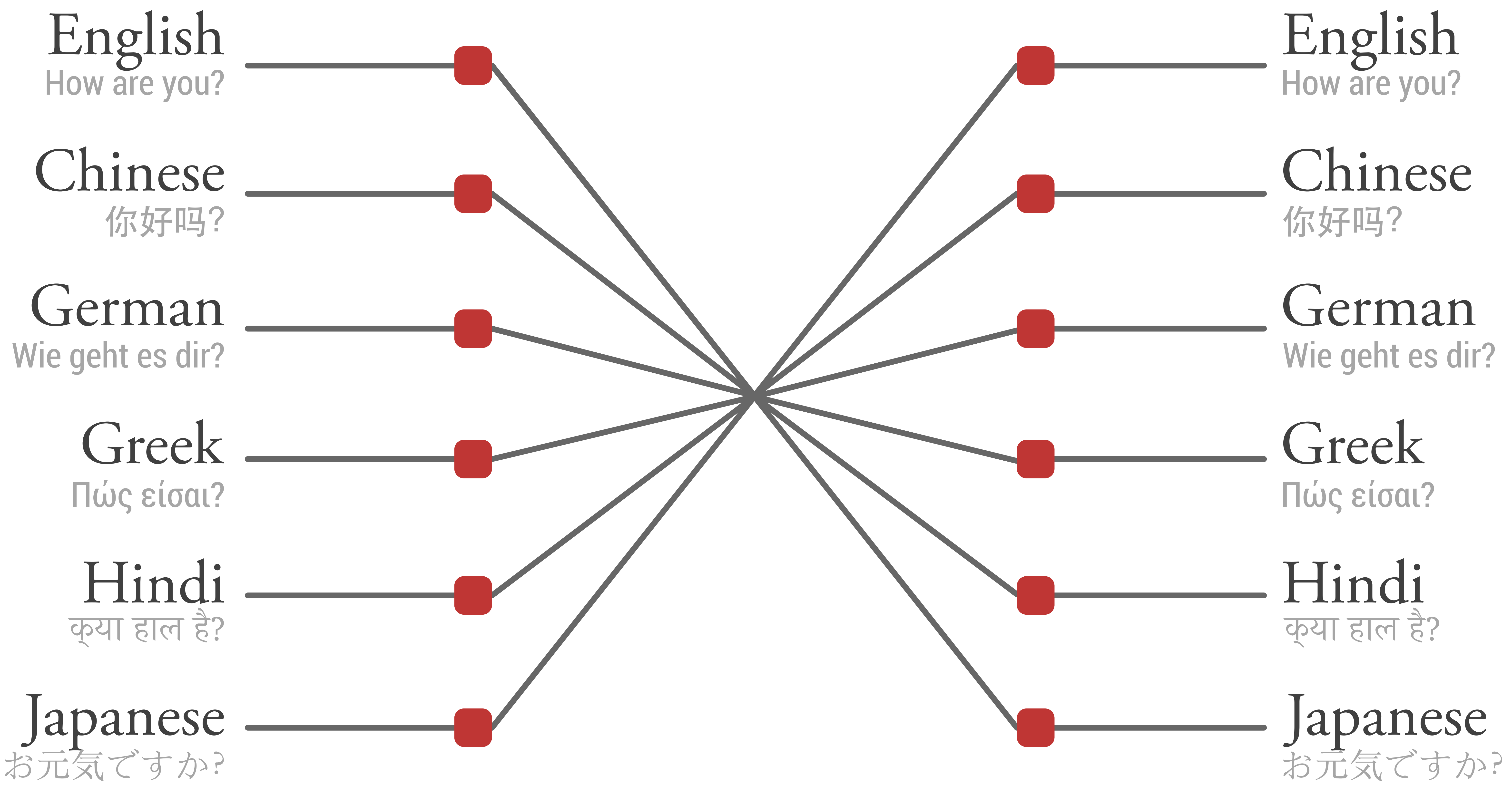


No Parameter Sharing

Contextual Parameter Generation

What about *multilingual translation*?





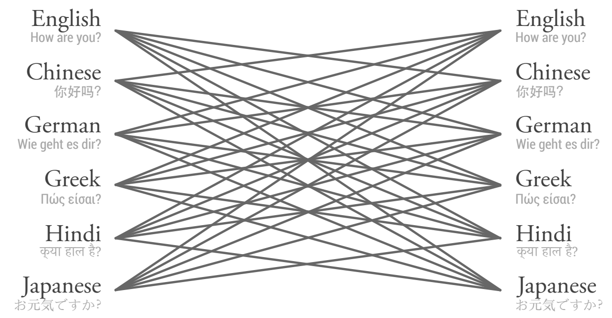
Hard Parameter Sharing

[Firat, Cho, Bengio, NAACL 2016 & Luong, Le, Sutskever, Vinyals, Kaizer, ICLR 2016]

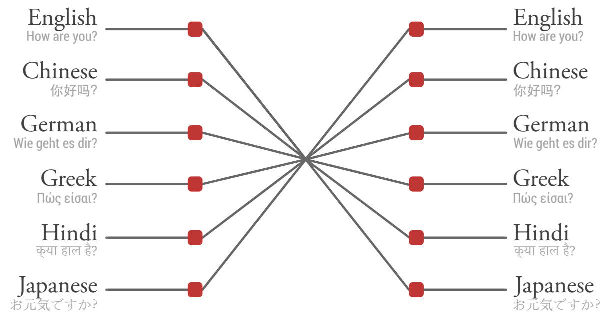
Contextual Parameter Generation

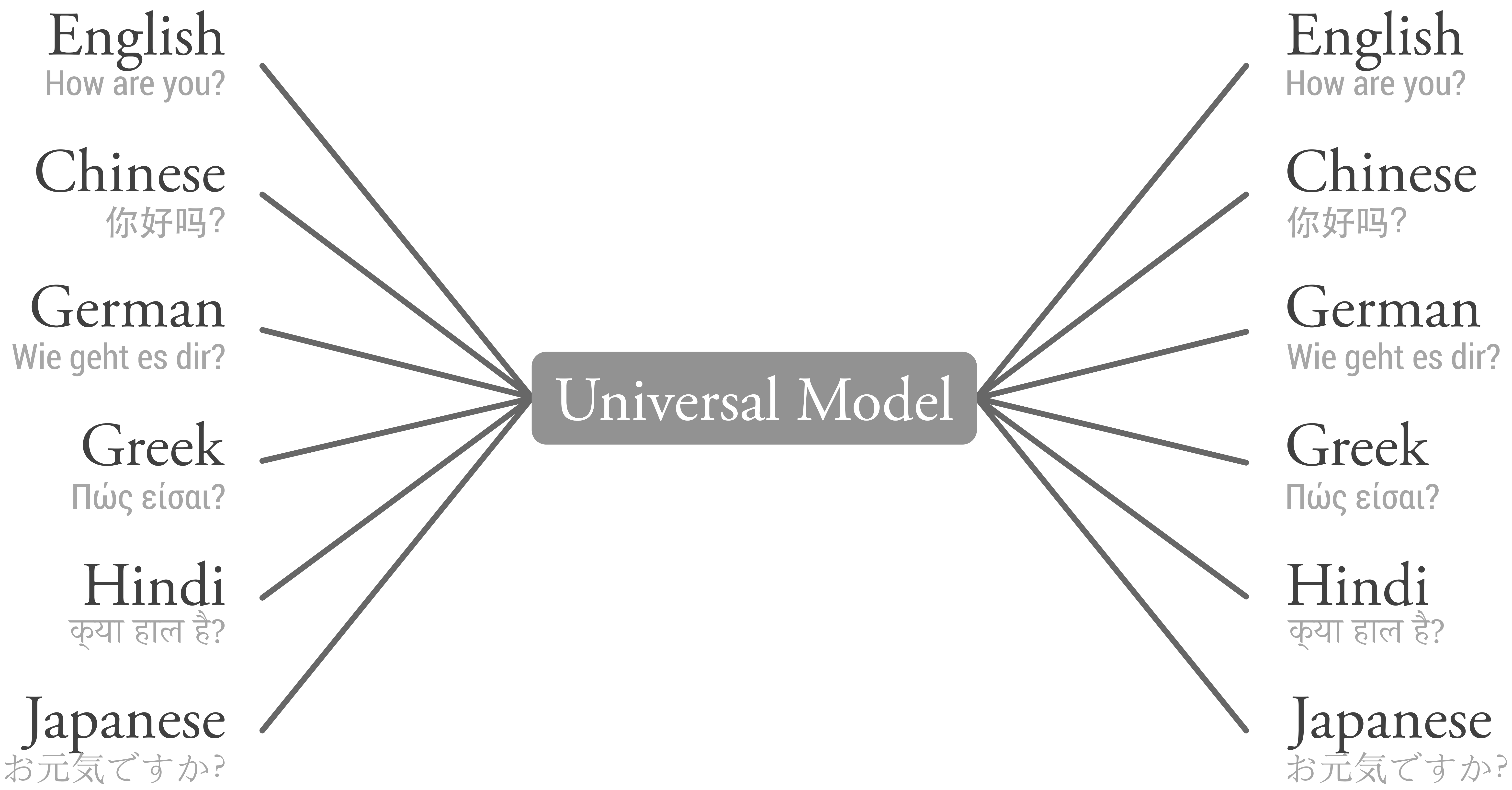
What about *multilingual translation*?

Pairwise



Per-Language Encoder/Decoder

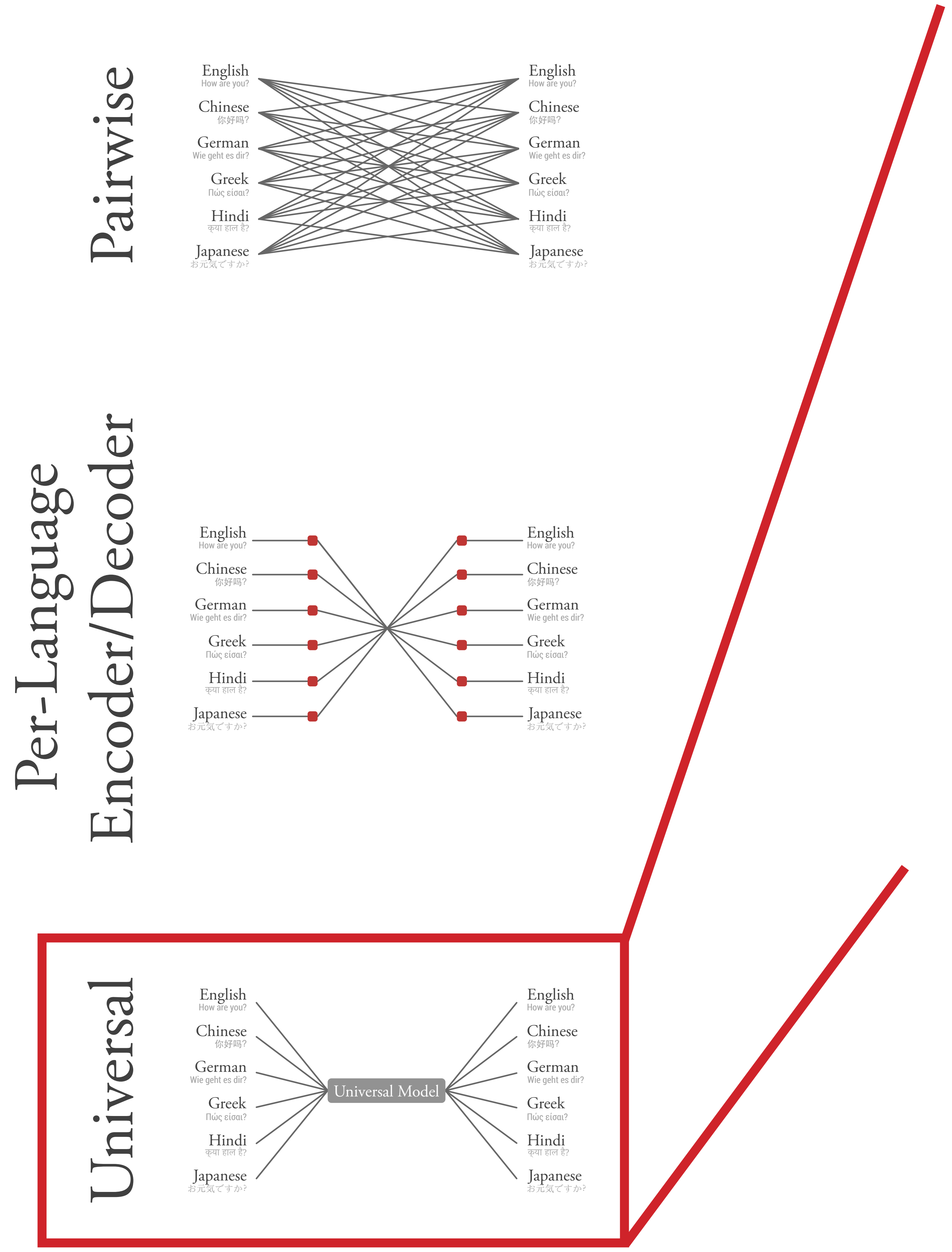


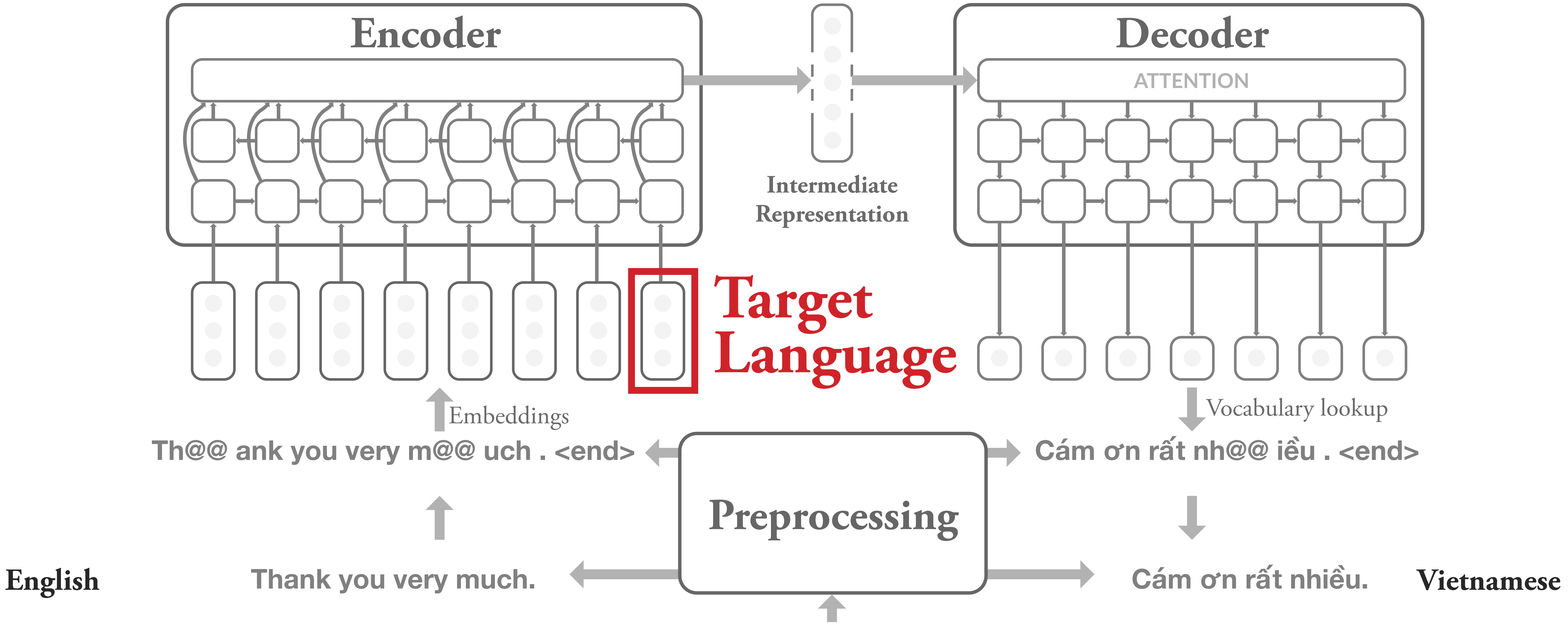


[Ha et al., IWSLT 2016 & Johnson et al., ACL 2017]

Contextual Parameter Generation

What about *multilingual translation*?





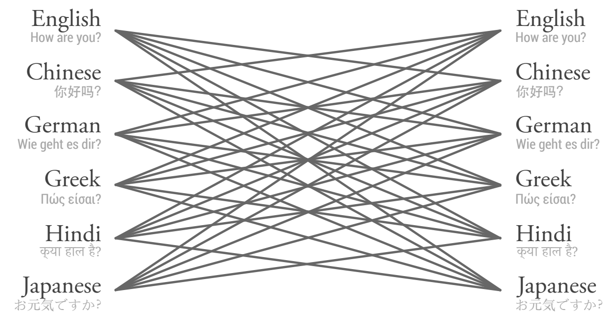
- Lacks language-specific parameterization.
- Cannot generalize to the other settings.
- It does not make sense to treat the target language in the same way as the input sentence words.

[Ha et al., IWSLT 2016 & Johnson et al., ACL 2017]

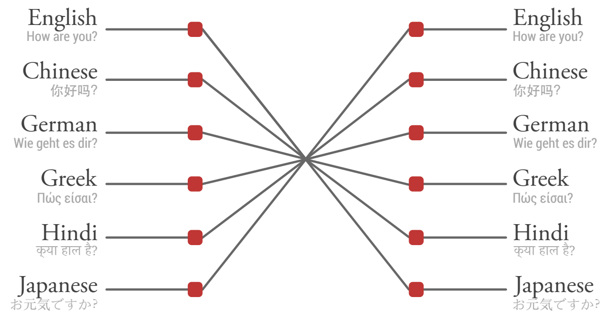
Contextual Parameter Generation

What about *multilingual translation*?

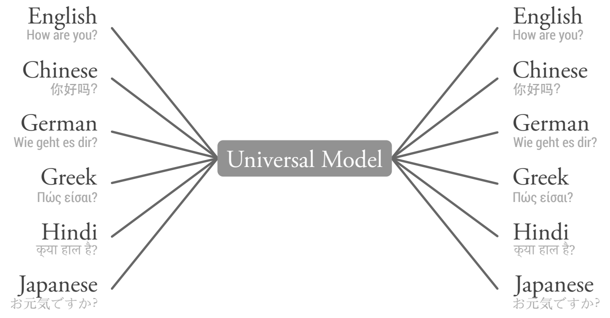
Pairwise



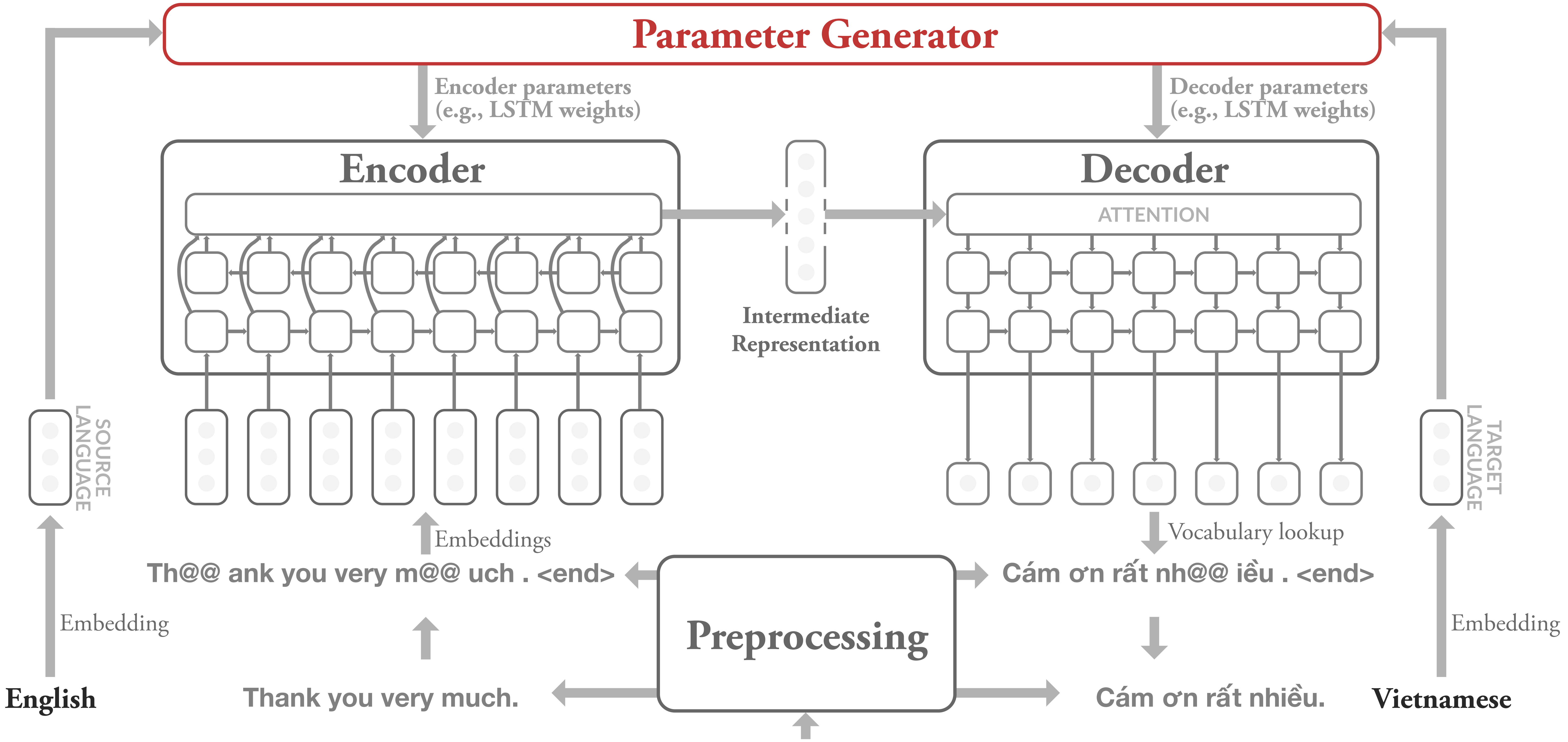
Per-Language Encoder/Decoder



Universal



We proposed to explicitly allow conditioning on the languages:



[**Platanios**, Sachan, Neubig, Mitchell, EMNLP 2018]

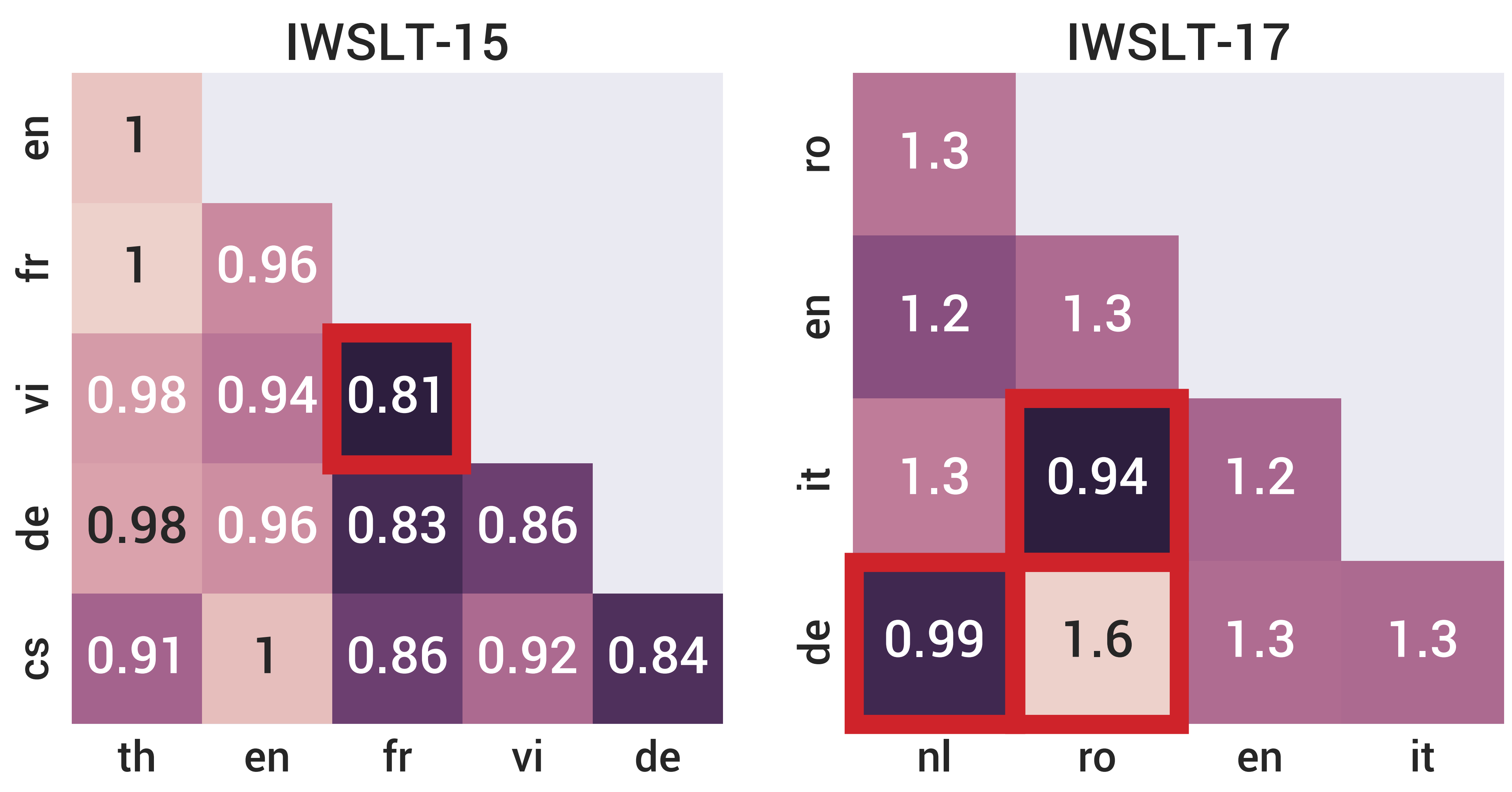
Contextual Parameter Generation

Language acts as the *context* in which translation is performed.

Using contextual parameter generation resulted in:

- Significant *performance gains* (+3 BLEU) and *reduced training time*.
- *Interpretable* task embeddings.

Cosine Distances:



[**Platanios**, Sachan, Neubig, Mitchell, EMNLP 2018]

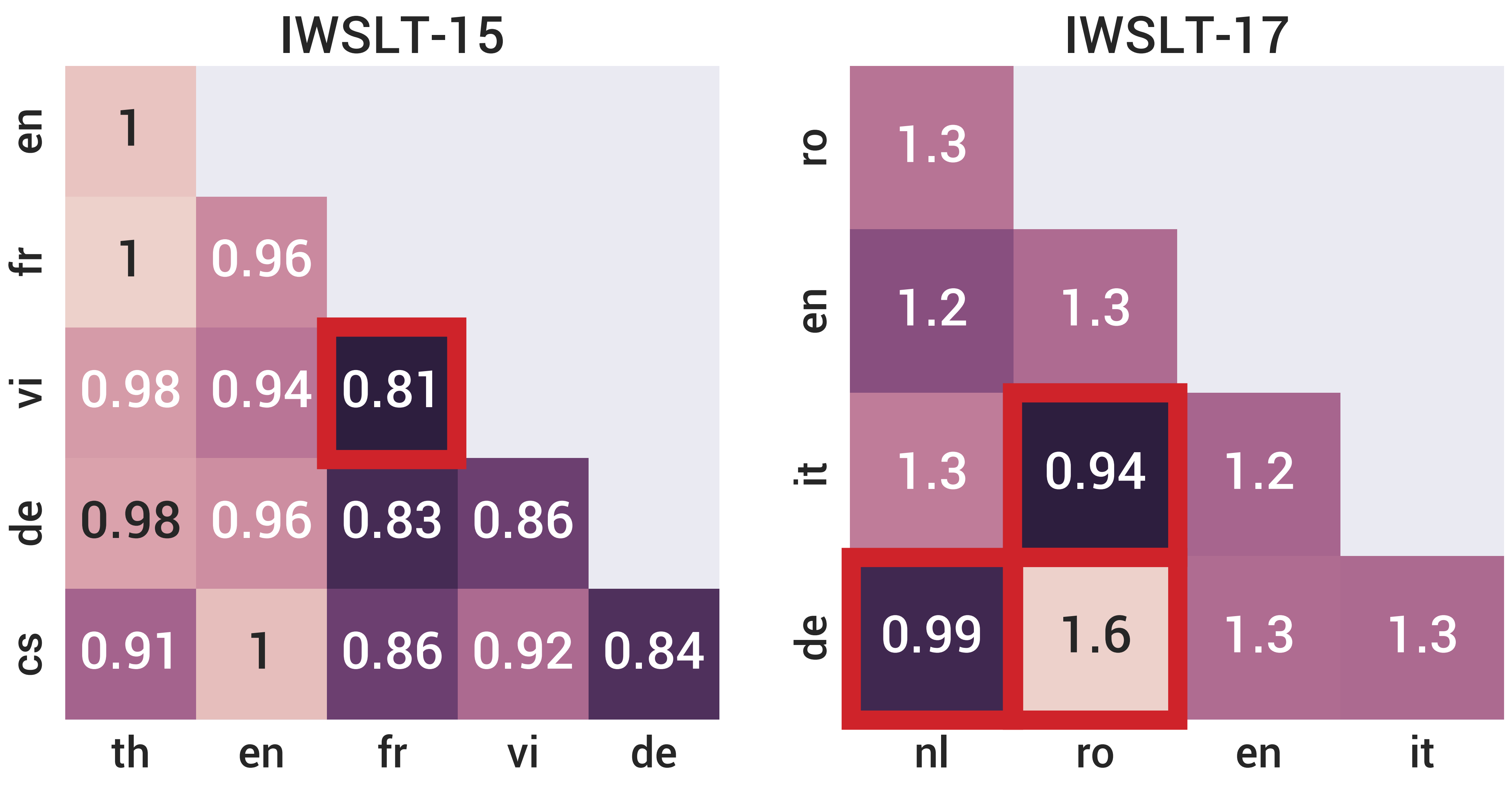
Contextual Parameter Generation

Language acts as the *context* in which translation is performed.

Using contextual parameter generation resulted in:

- Significant *performance gains* (+3 BLEU) and *reduced training time*.
- *Interpretable* task embeddings.

We also have positive preliminary results for knowledge graph link prediction and other tasks!

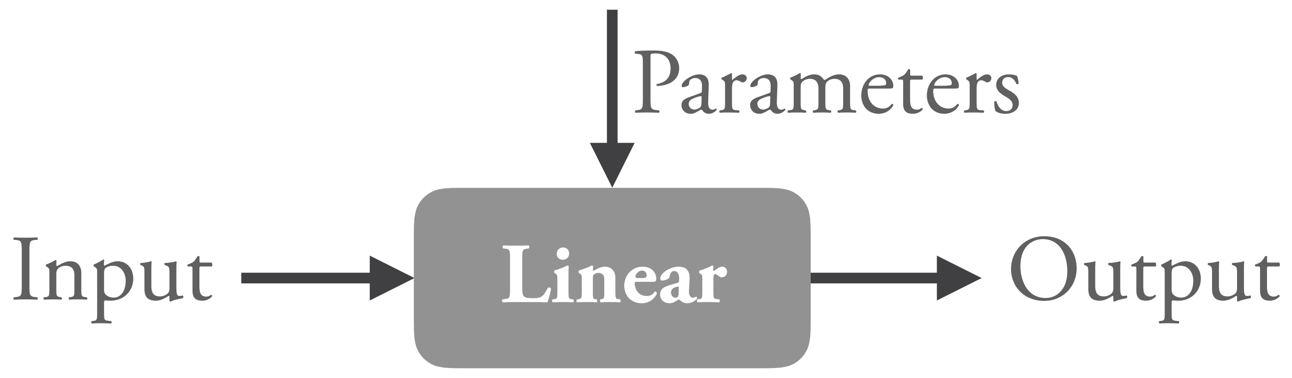


[**Platanios**, Sachan, Neubig, Mitchell, EMNLP 2018]

Contextual Parameter Generation

It remains to answer some questions:

- Why does contextual parameter generation work so well?
- Is contextual parameter generation related to probabilistic graphical models?
- How does it increase the expressive power of neural networks?

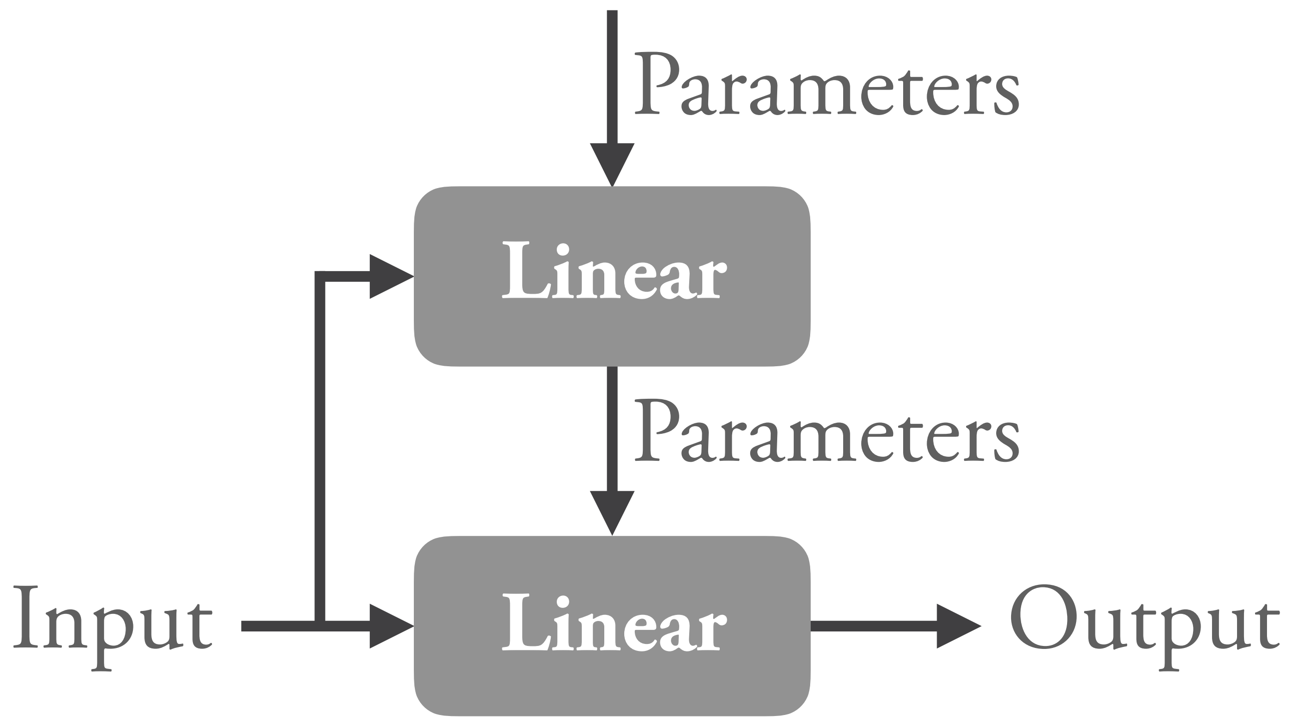


Cannot represent the XOR function!

Contextual Parameter Generation

It remains to answer some questions:

- Why does contextual parameter generation work so well?
- Is contextual parameter generation related to probabilistic graphical models?
- How does it increase the expressive power of neural networks?

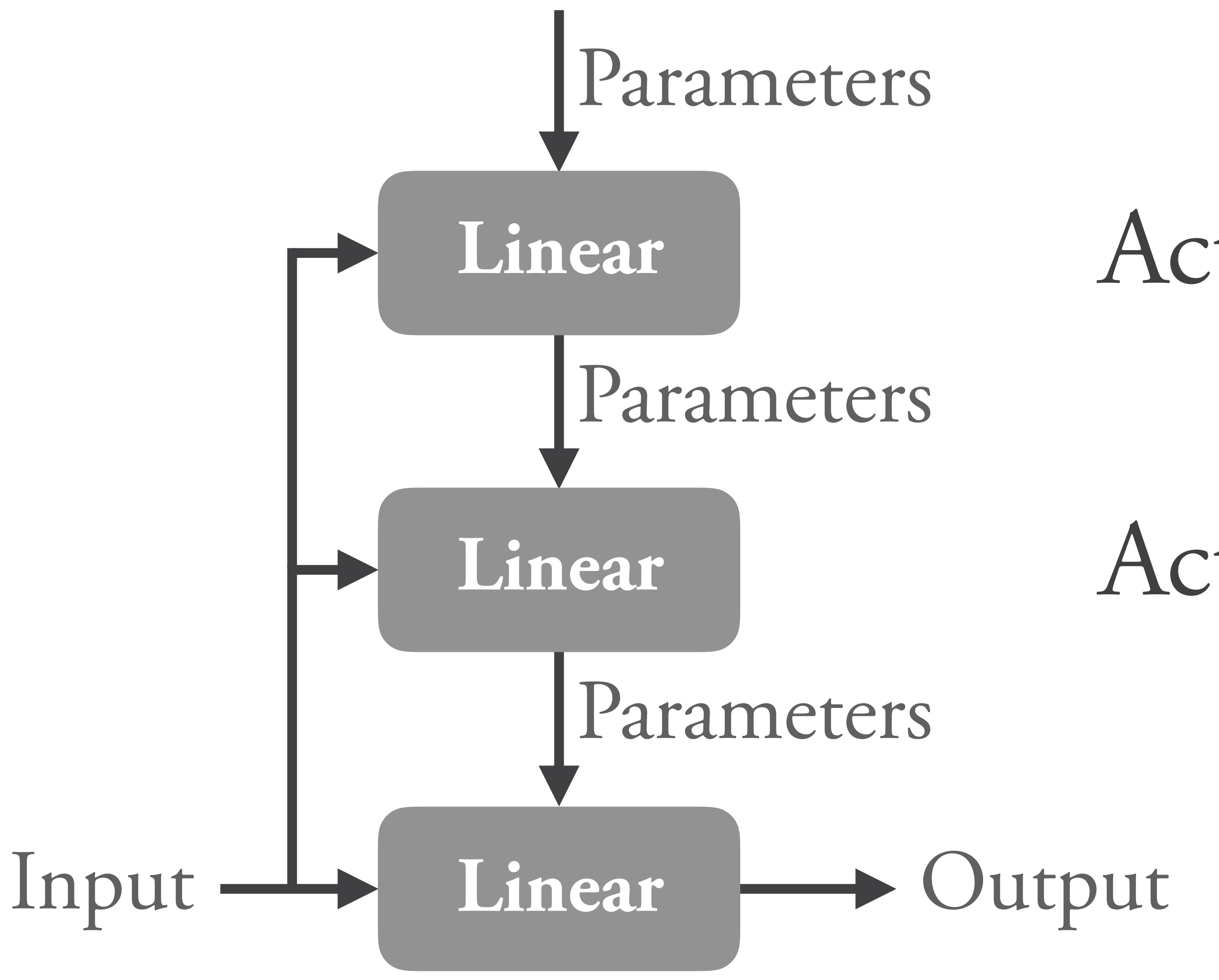


Can represent the XOR function!

Contextual Parameter Generation

It remains to answer some questions:

- Why does contextual parameter generation work so well?
- Is contextual parameter generation related to probabilistic graphical models?
- How does it increase the expressive power of neural networks?



Acts a **cubic kernel!**

Acts a **quadratic kernel!**

Can represent the XOR function!

Outline

Part 1: Proposed Approach

1. Learning from Noisy Labels
   supervision noise and ambiguity
2. Contextual Parameter Generation
   multi-task learning
3. Self-Reflection
   unsupervised learning
4. Unified Architecture
   general learning and intelligence

Part 2: Evaluation

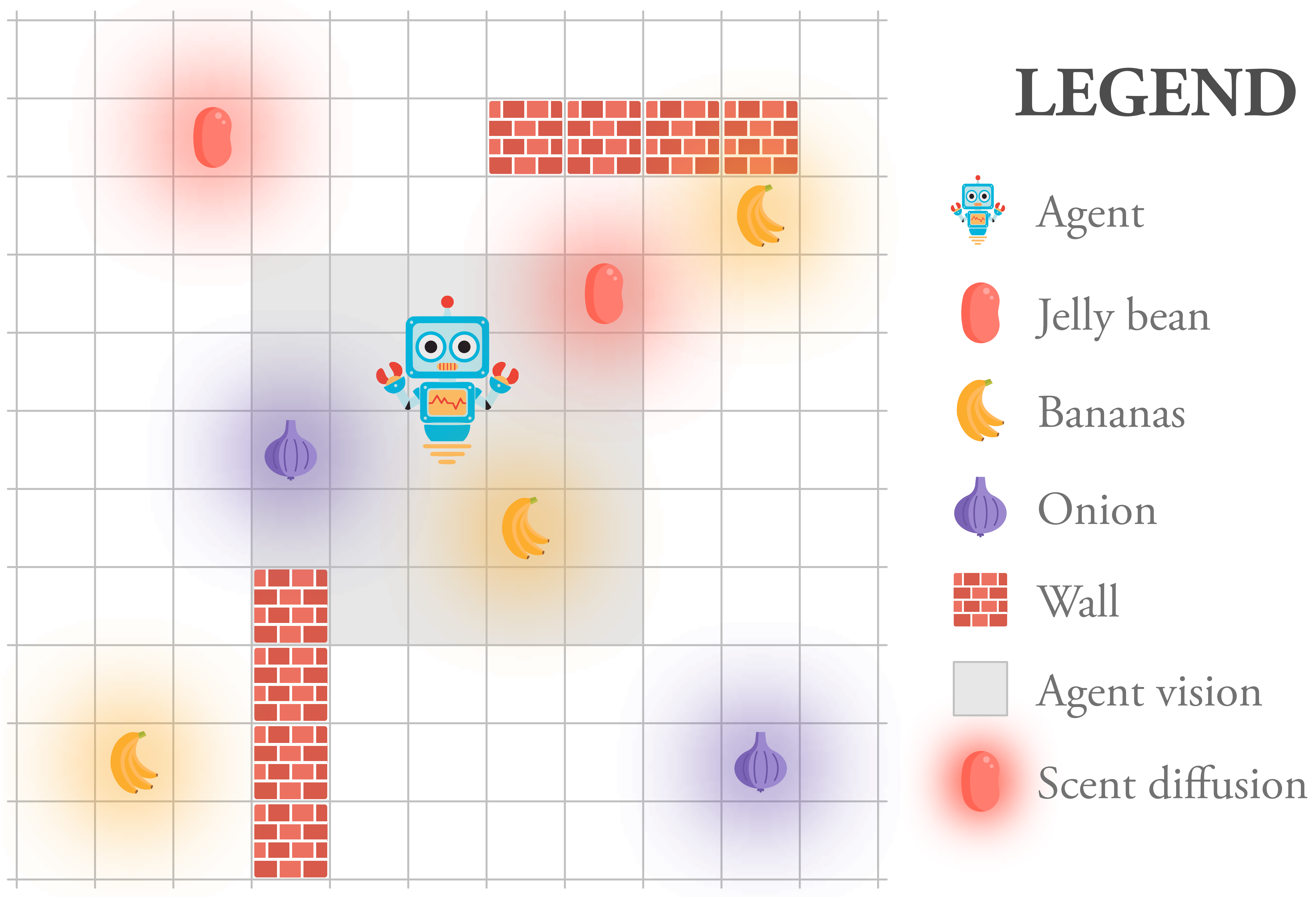
Part 3: Timeline

Self-Reflection

Let us consider the following example:

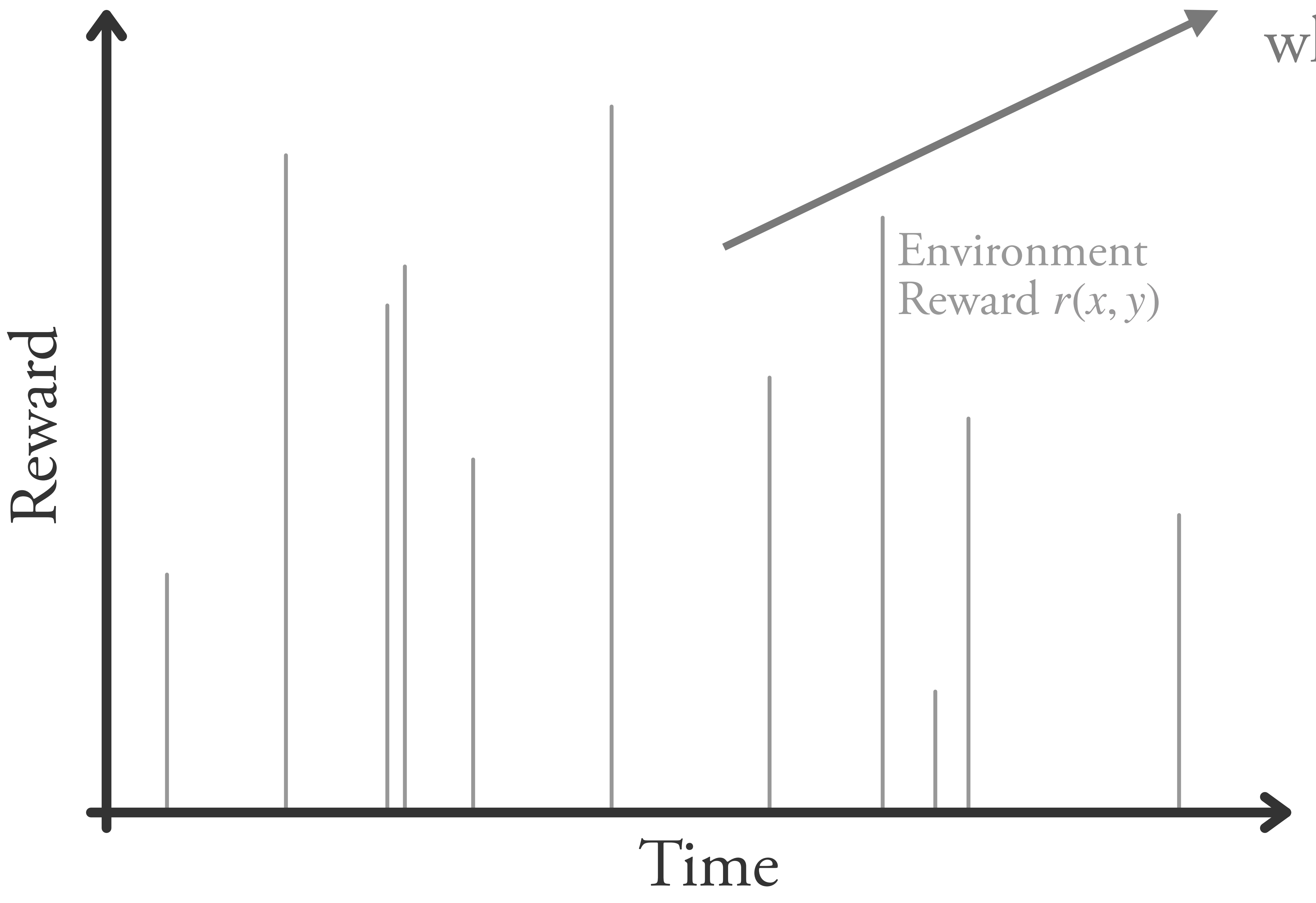
- Procedurally Generated
- Multi-Modal:
  - **Vision:** high precision / low recall
  - **Scent:** low precision / high recall

Collect 5 jelly beans and 2 bananas!



Self-Reflection

Never-ending learning requires that we leverage *unlabeled* data. How can we do that?



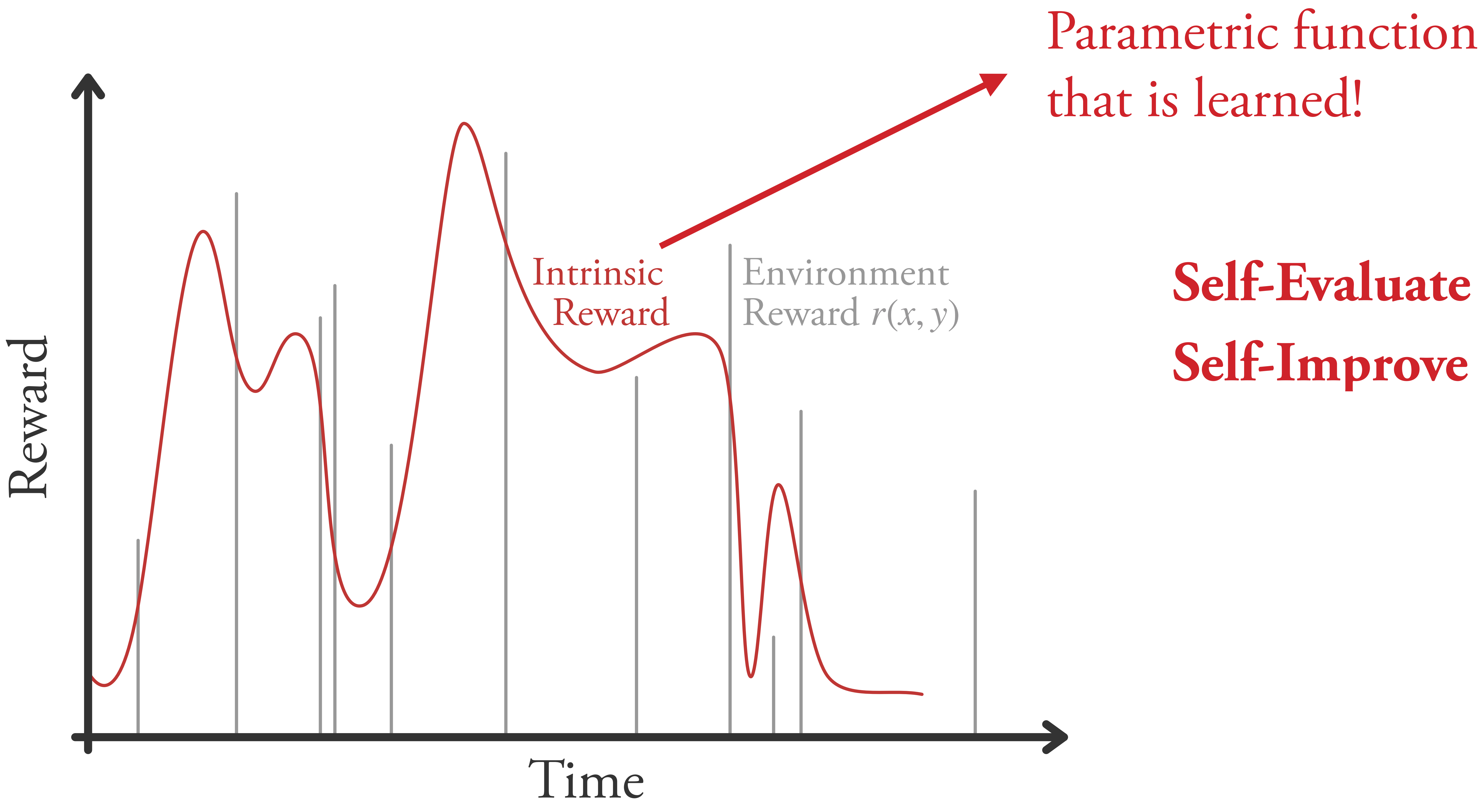
What would a human do when there is no reward?

Self-Evaluate

Self-Improve

Self-Reflection

Never-ending learning requires that we leverage *unlabeled* data. How can we do that?



Self-Evaluate

Self-Improve

Outline

Part 1: Proposed Approach

1. Learning from Noisy Labels
   supervision noise and ambiguity
2. Contextual Parameter Generation
   multi-task learning
3. Self-Reflection
   unsupervised learning
4. Unified Architecture
   general learning and intelligence

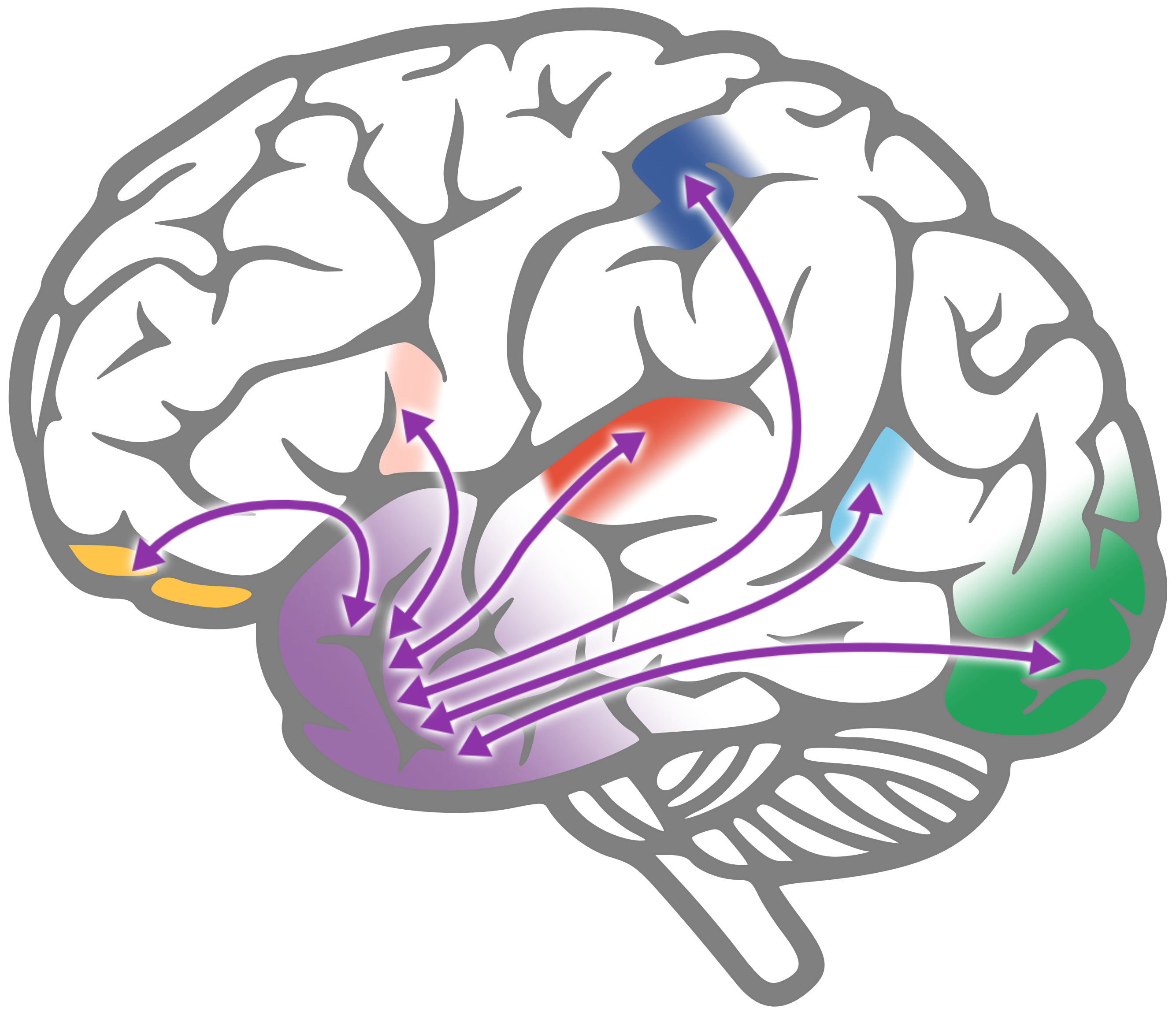
Part 2: Evaluation

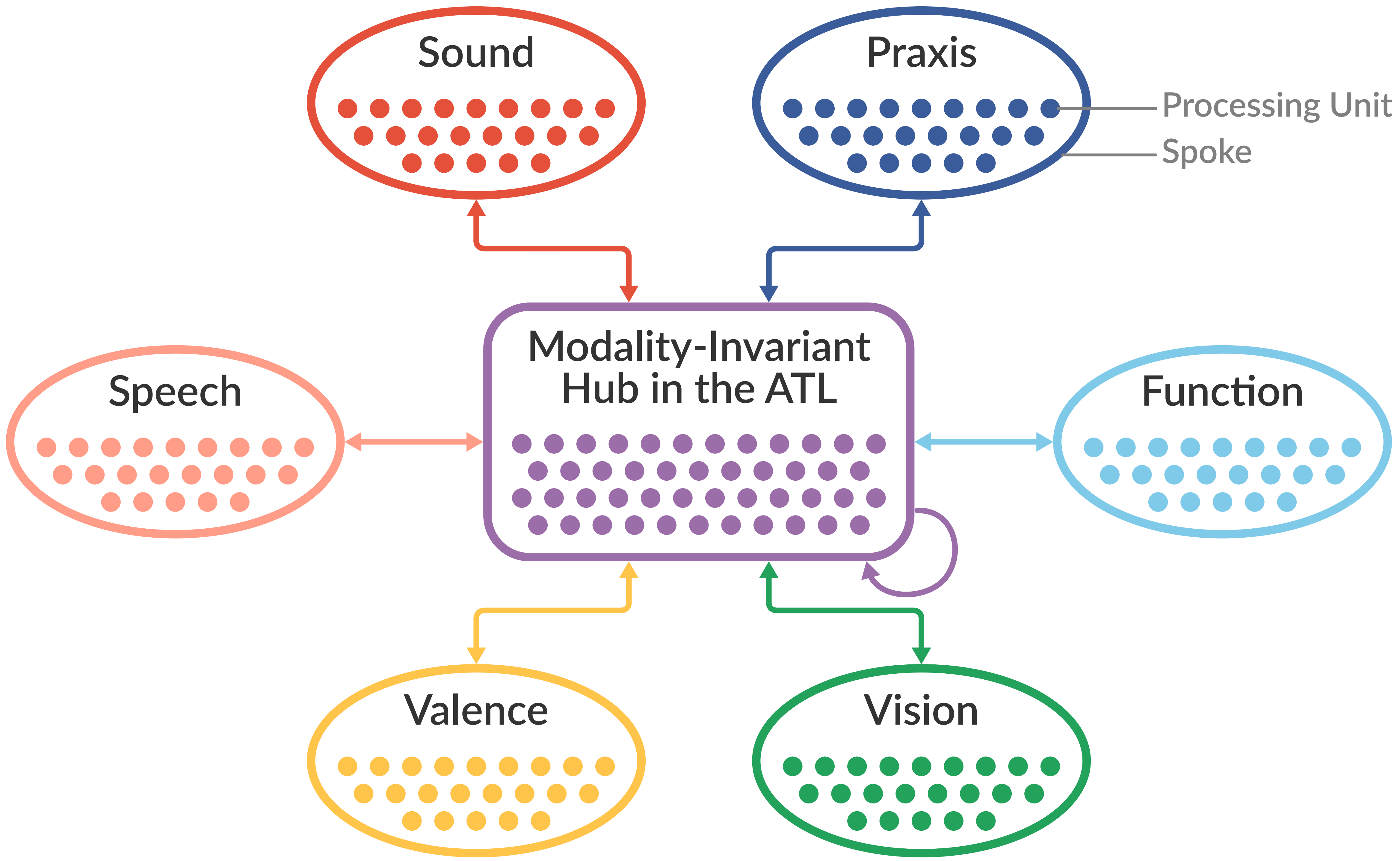
Part 3: Timeline

Unified Architecture

How do all the pieces fit together in the *human brain*?

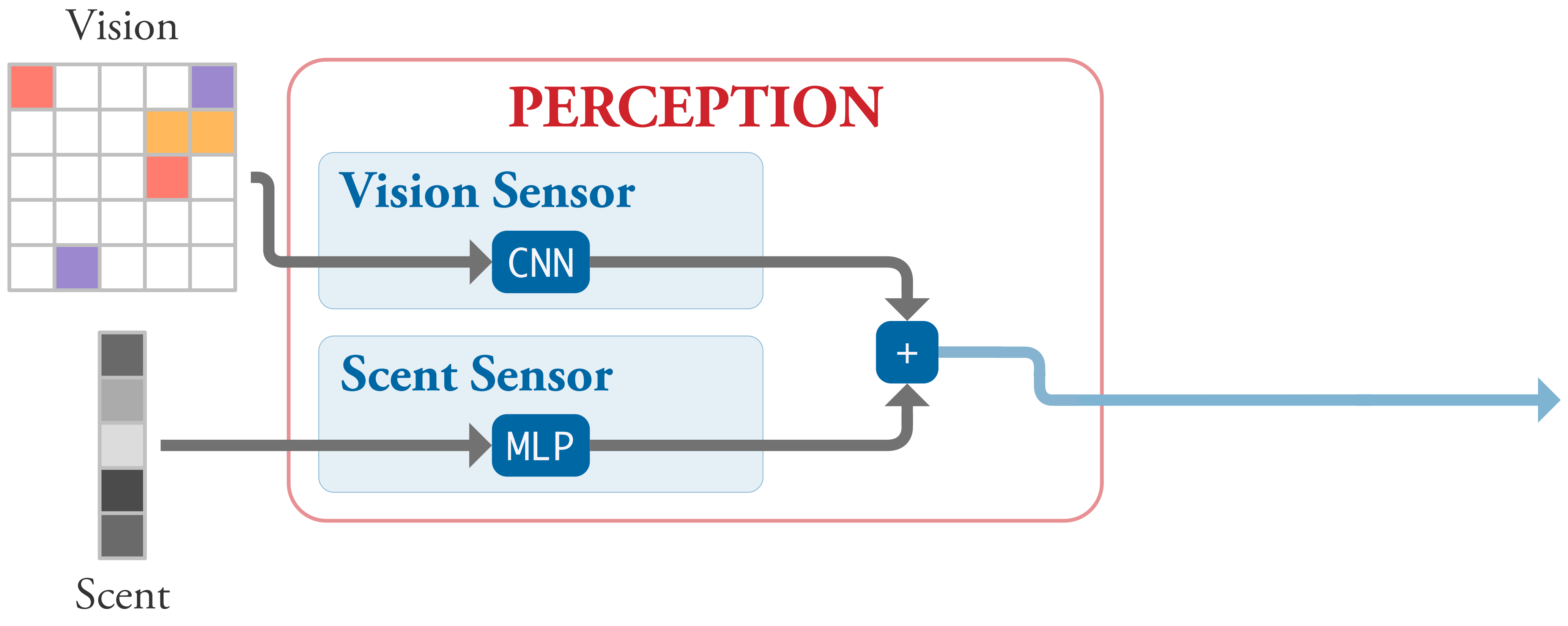
The Hub-and-Spoke Model



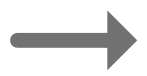


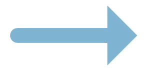
[Rogers et al., Psychological Review 2004]

Unified Architecture: **JBW Example**



LEGEND


Flow of data whose type is modality/problem dependent


Flow of data whose type is modality/problem independent

Unified Architecture

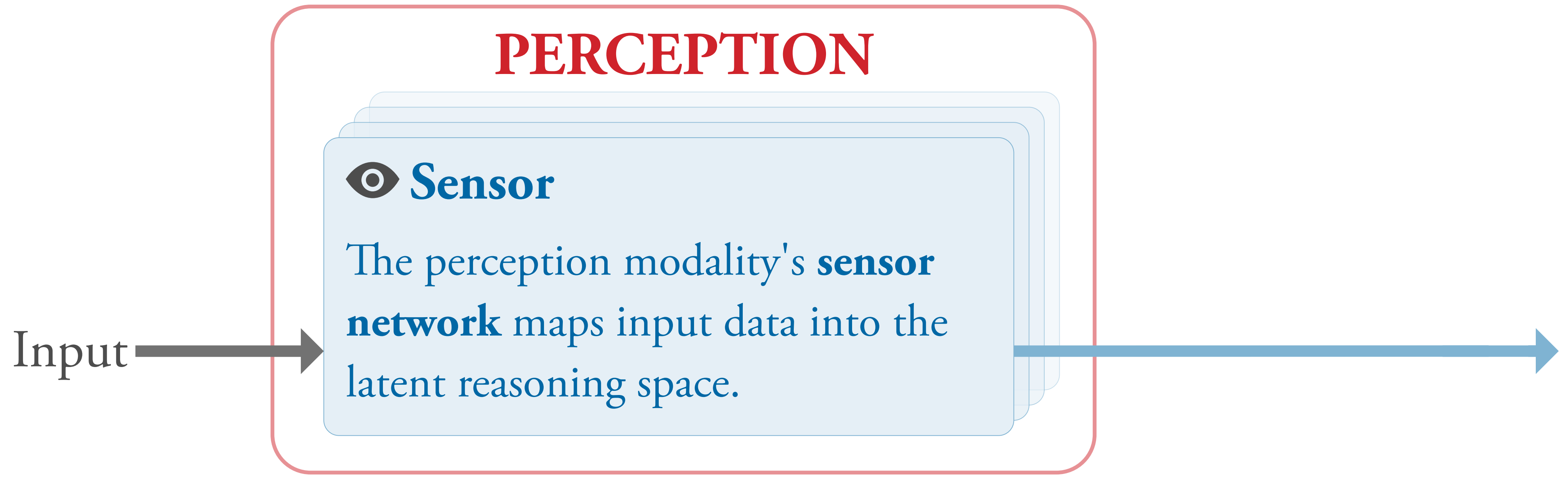
A **perception modality** is defined as:

- A *data type*, and
- a *sensor network*.

LEGEND

Flow of data whose type is modality/problem dependent

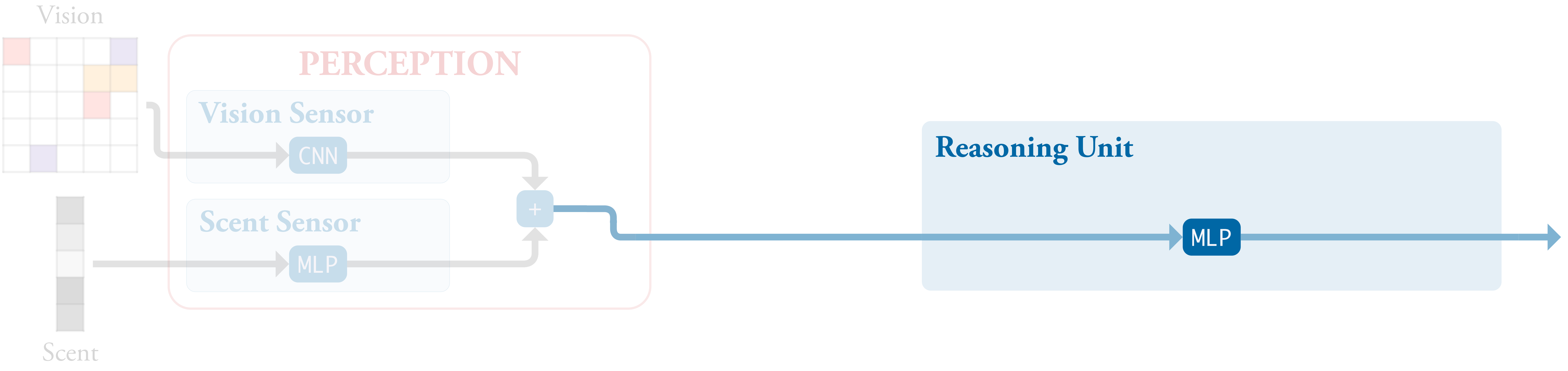
Flow of data whose type is modality/problem independent



Unified Architecture: **JBW Example**

LEGEND

- Flow of data whose type is modality/problem dependent
- Flow of data whose type is modality/problem independent

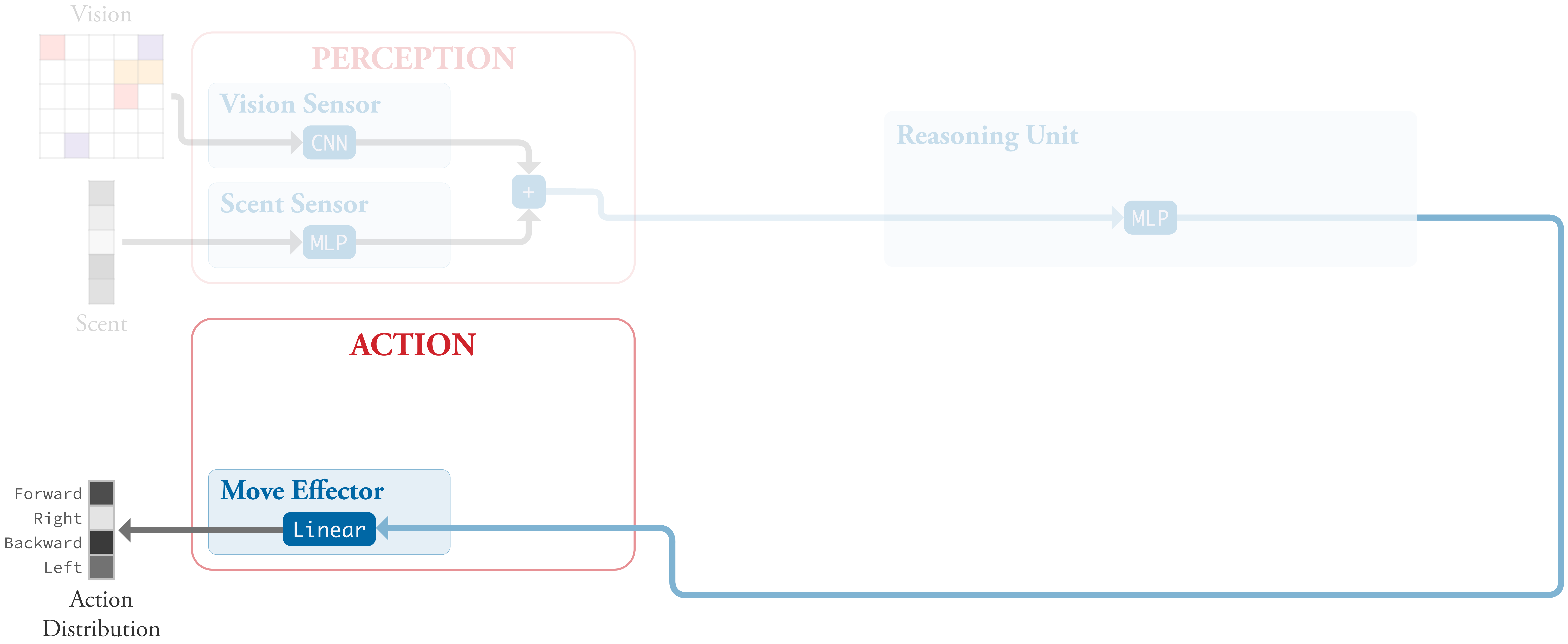


Much of the complexity of deep learning models lies in perception, rather than reasoning.

Unified Architecture: JBW Example

LEGEND

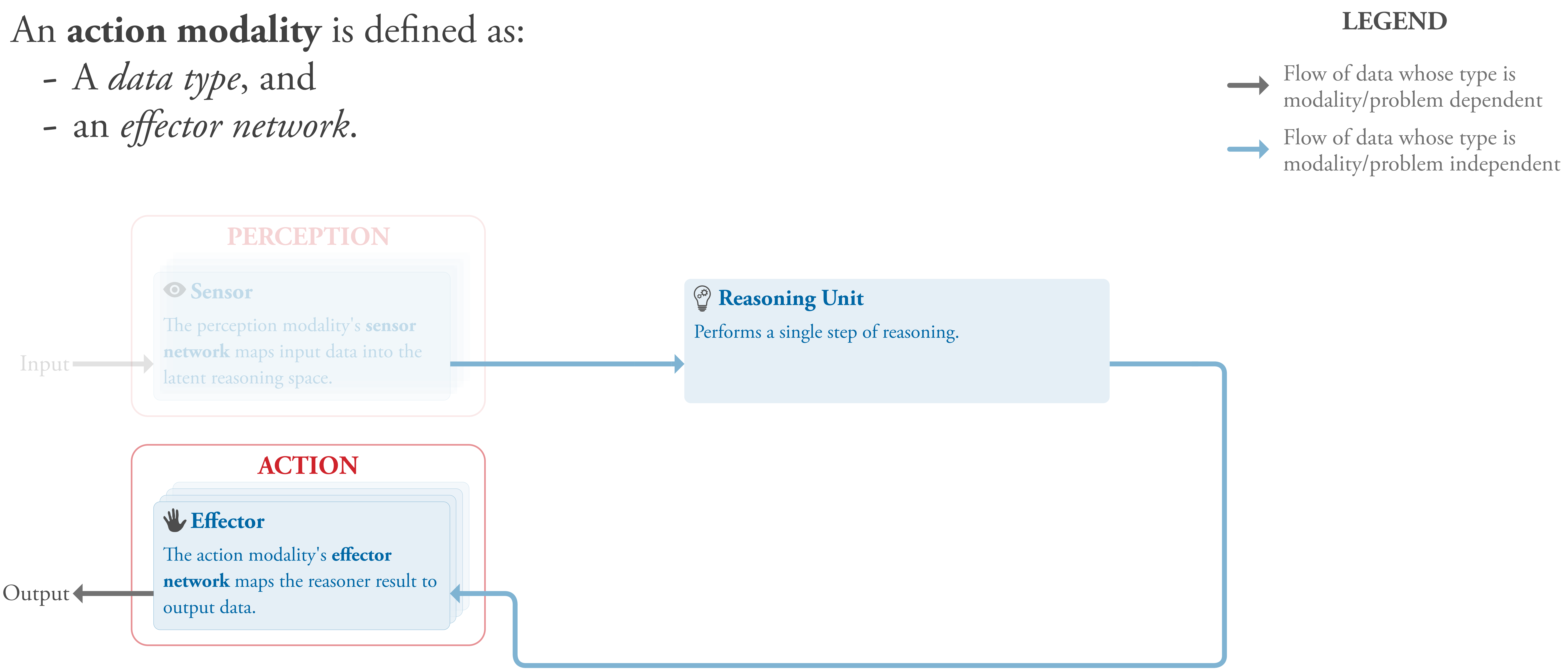
- Flow of data whose type is modality/problem dependent
- Flow of data whose type is modality/problem independent



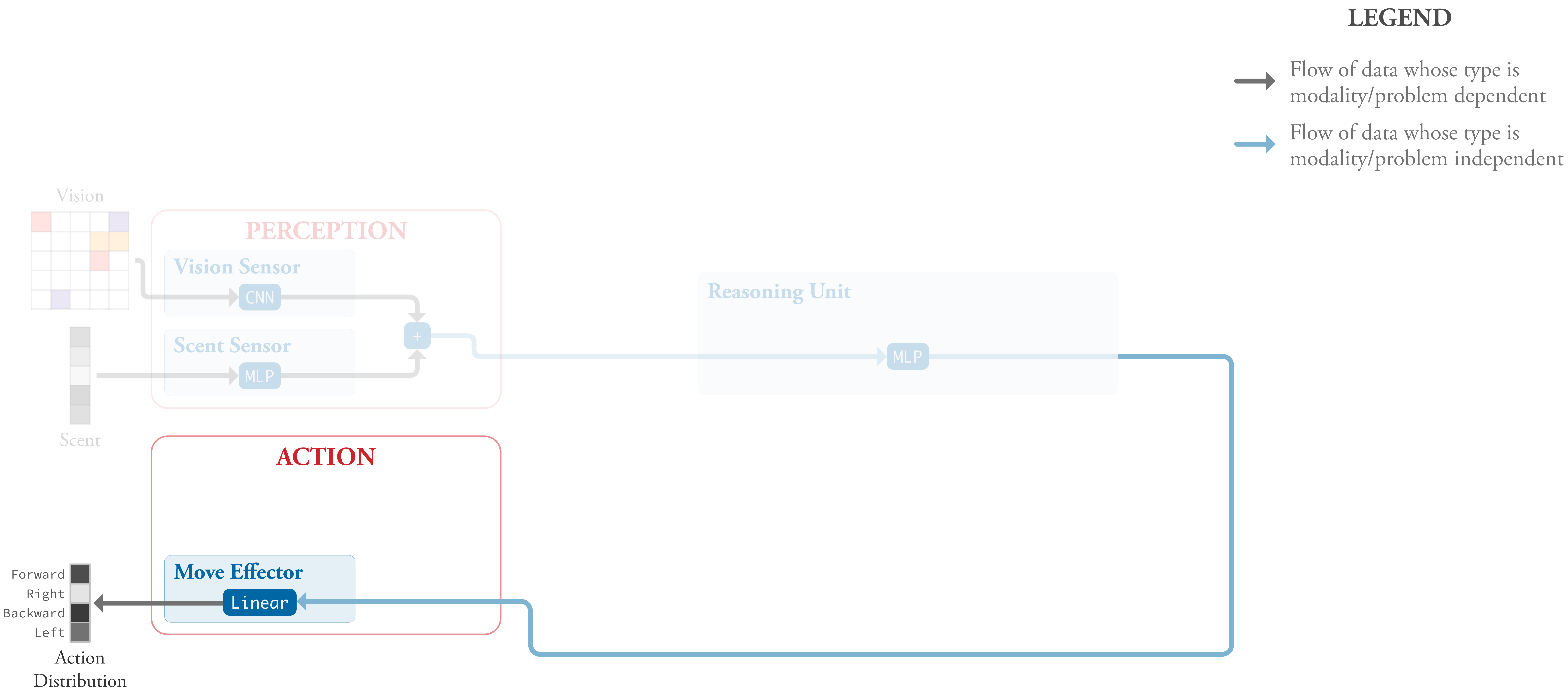
Unified Architecture

An **action modality** is defined as:

- A *data type*, and
- an *effector network*.



Unified Architecture: JBW Example

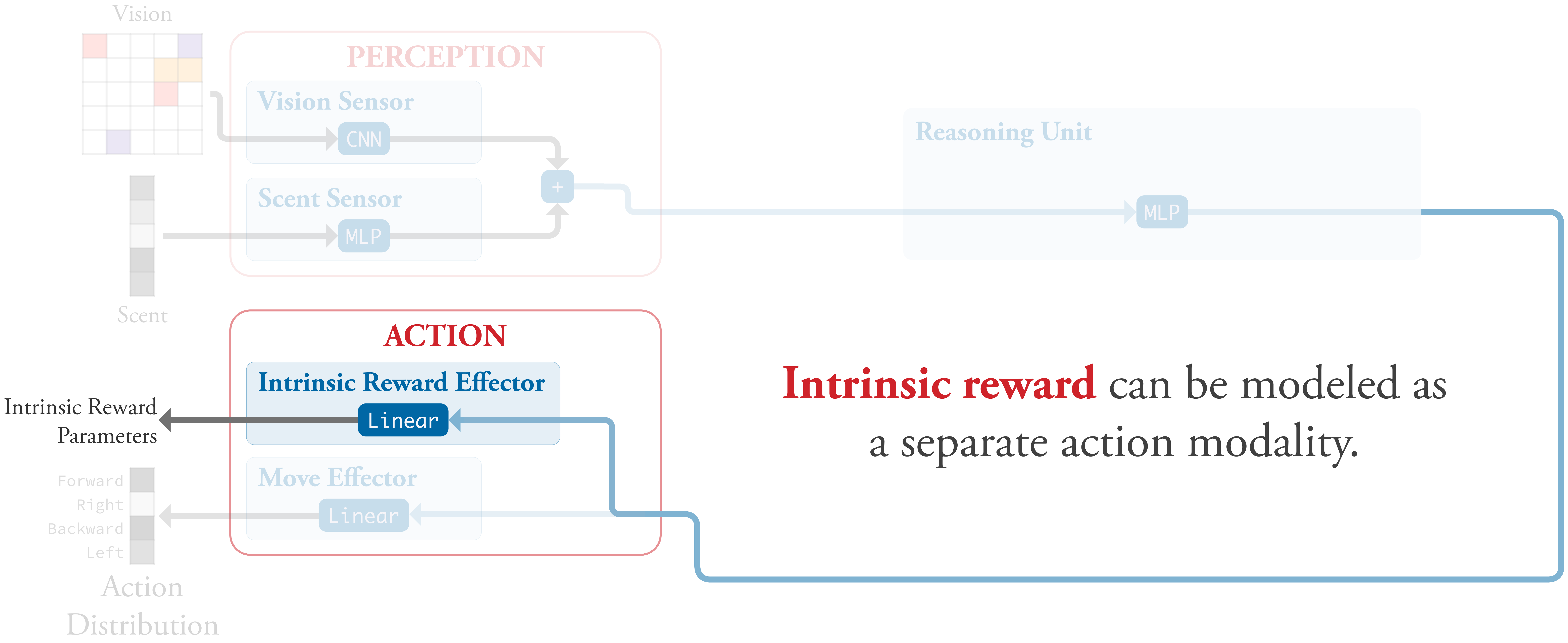


Unified Architecture: **JBW Example**

LEGEND

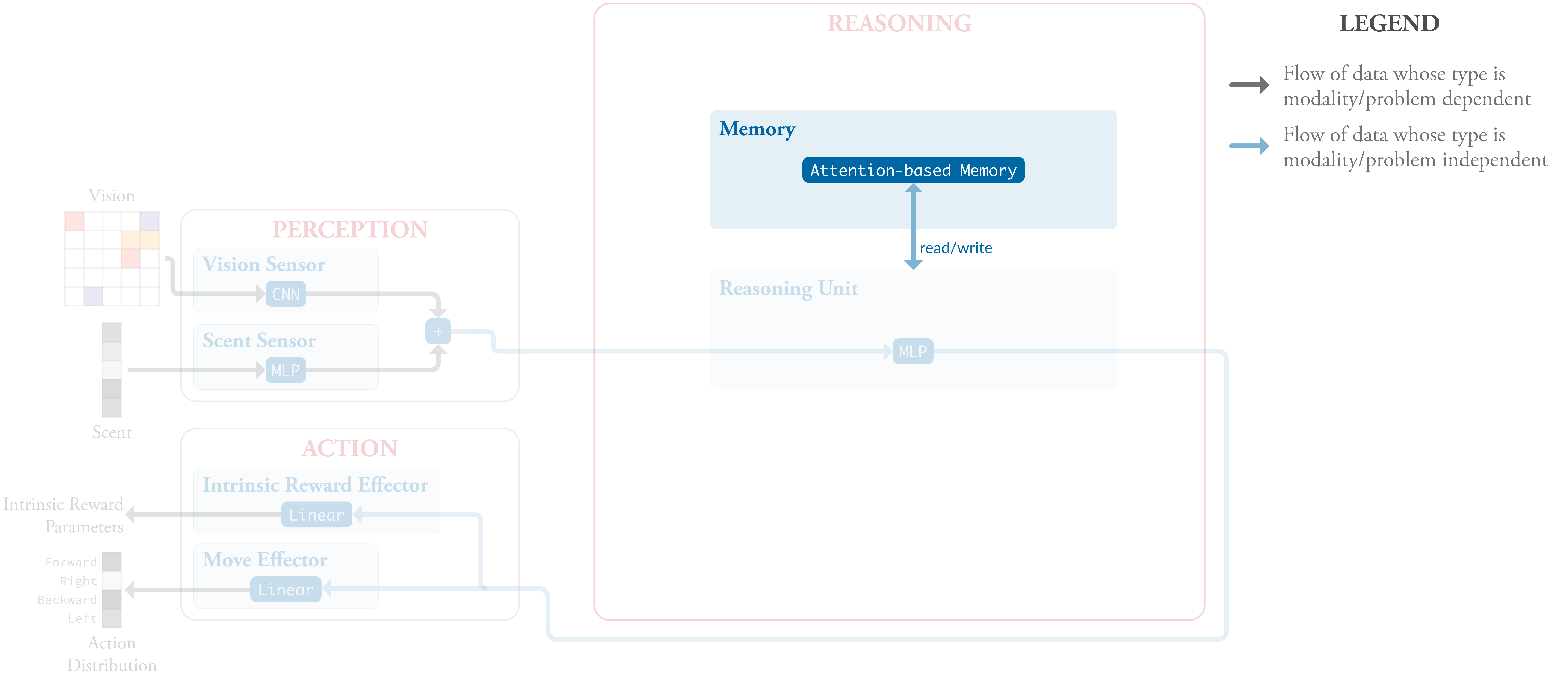
Flow of data whose type is modality/problem dependent

Flow of data whose type is modality/problem independent

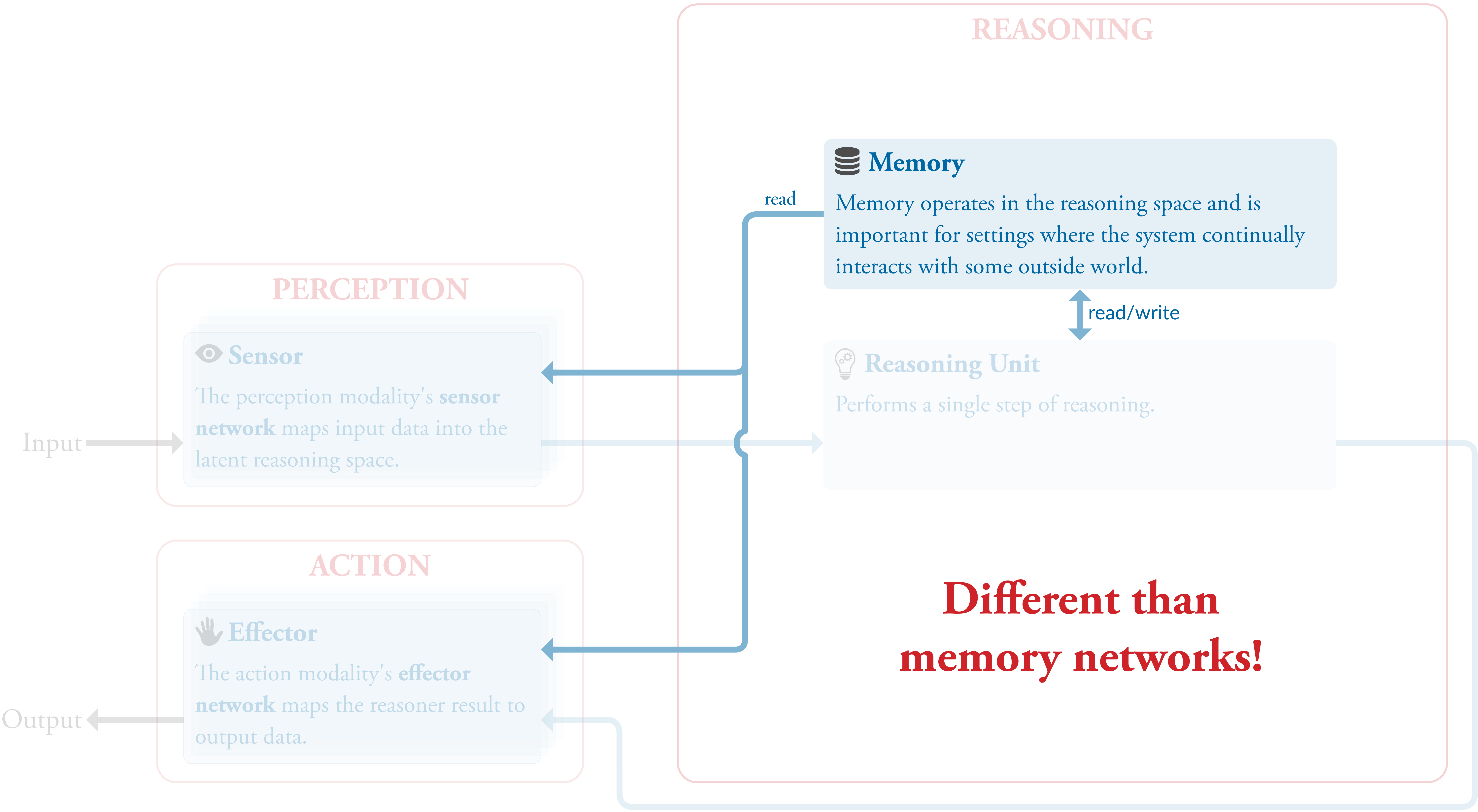


Intrinsic reward can be modeled as a separate action modality.

Unified Architecture: **JBW Example**



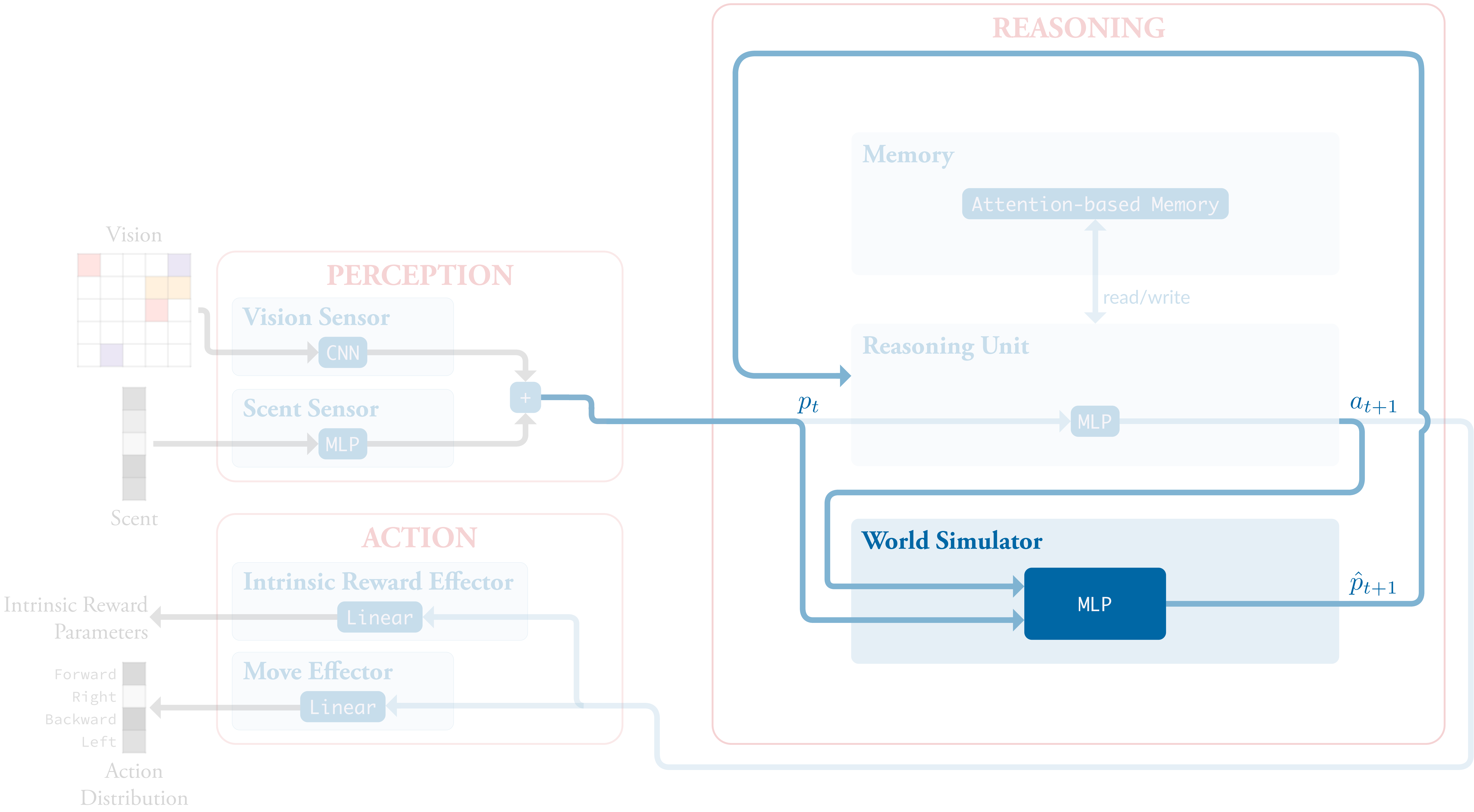
Unified Architecture



LEGEND

- Flow of data whose type is modality/problem dependent
- Flow of data whose type is modality/problem independent

Unified Architecture: JBW Example



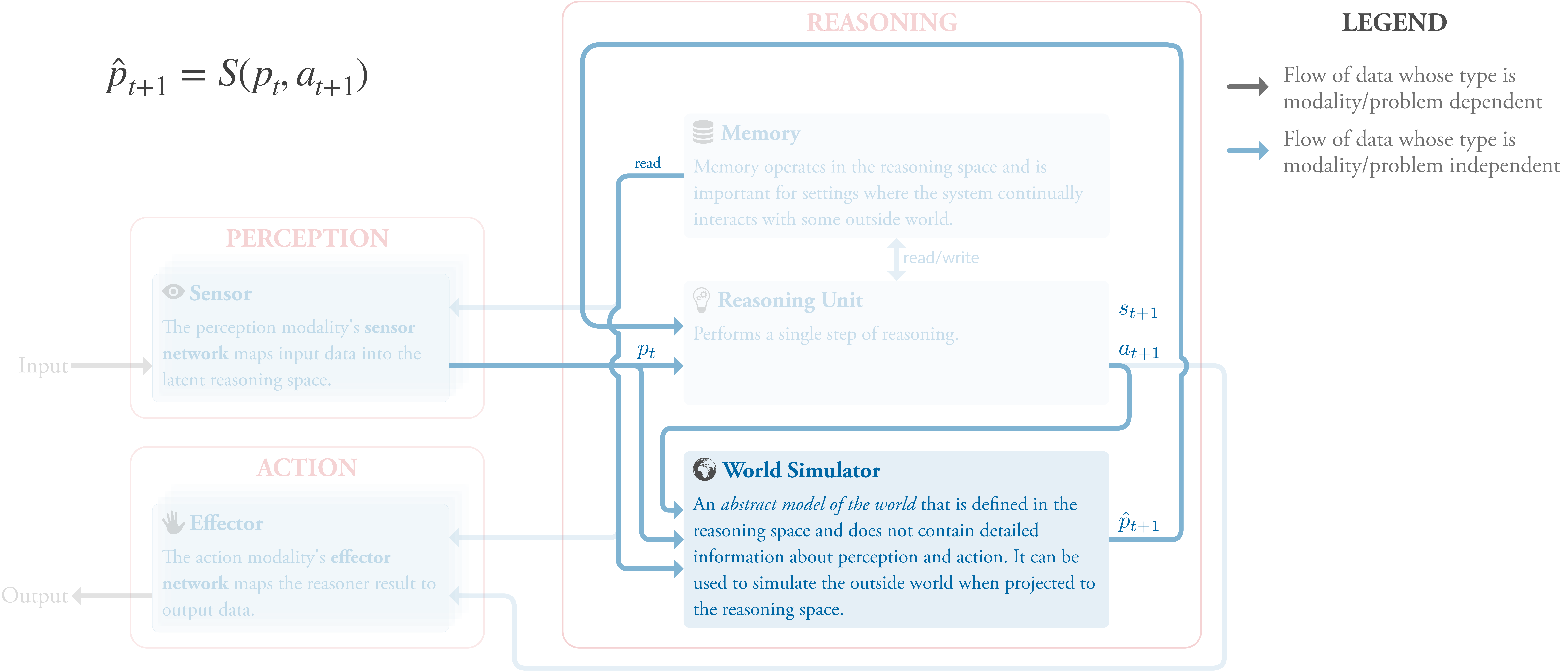
LEGEND

Flow of data whose type is modality/problem dependent

Flow of data whose type is modality/problem independent

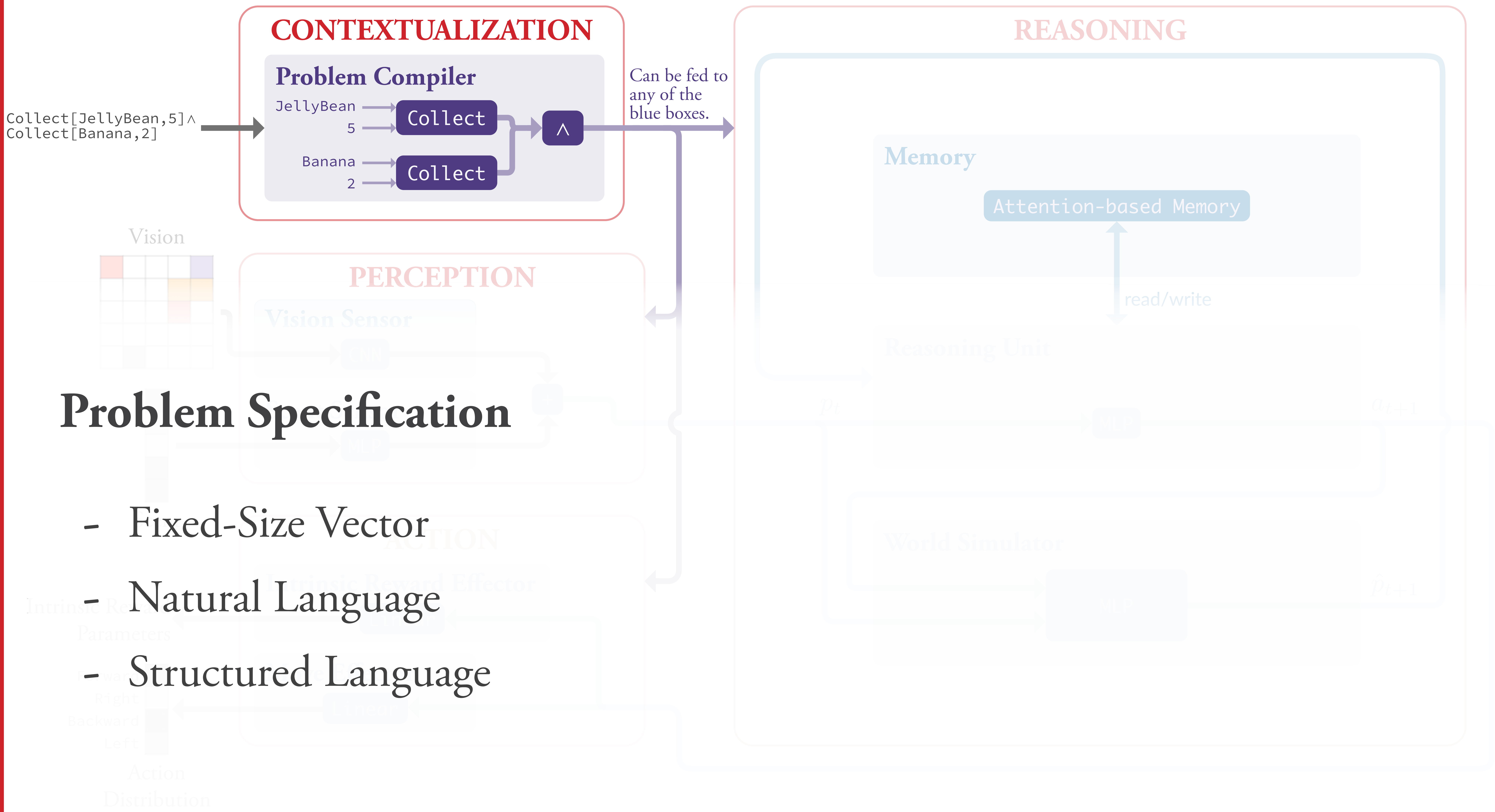
Unified Architecture

$$\hat{p}_{t+1} = S(p_t, a_{t+1})$$



Unified Architecture: JBW Example

Goal Contextualization



LEGEND

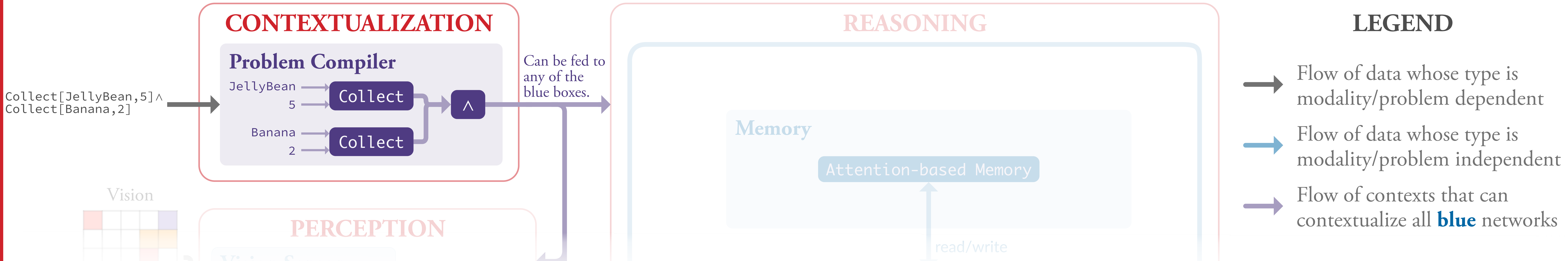
- Flow of data whose type is modality/problem dependent
- Flow of data whose type is modality/problem independent
- Flow of contexts that can contextualize all **blue** networks

Problem Specification

- Fixed-Size Vector
- Natural Language
- Structured Language

Unified Architecture: JBW Example

Goal Contextualization



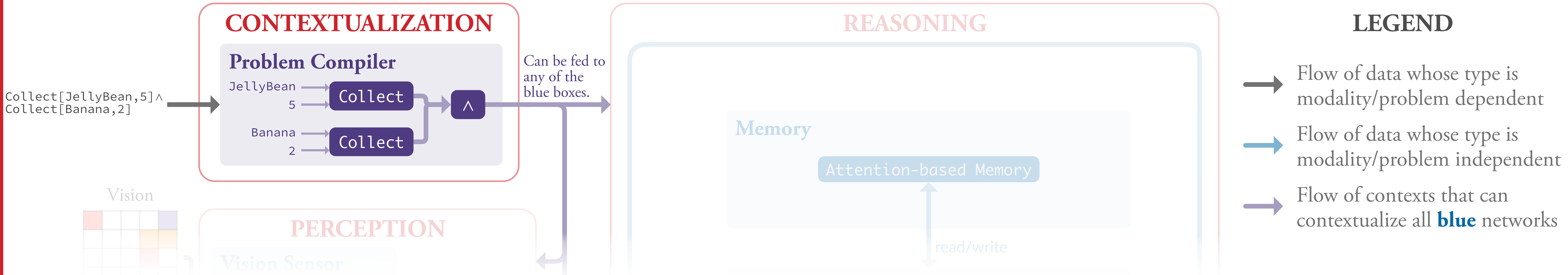
Problem Specification

- Fixed-Size Vector
- Natural Language
- Structured Language

Problem Compilation

Problem Specification → Structured Representation (e.g., a vector).

Unified Architecture: **JBW Example** Goal Contextualization



Problem Specification

- Fixed-Size Vector
- Natural Language
- Structured Language

Problem Compilation

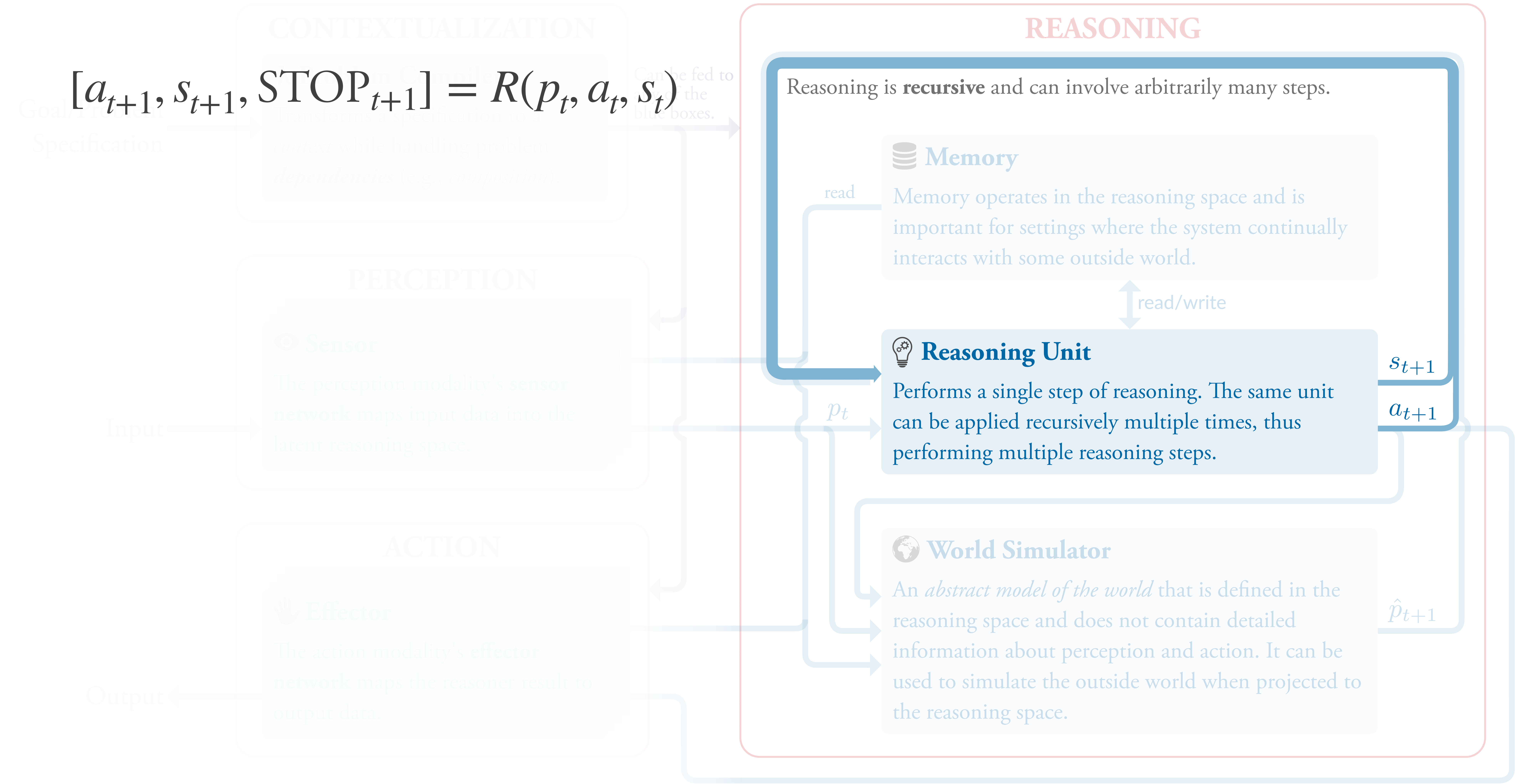
Problem Specification → Structured Representation (e.g., a vector).

Problem Generation

Action modalities can generate problems that are fed to the compiler.

Unified Architecture

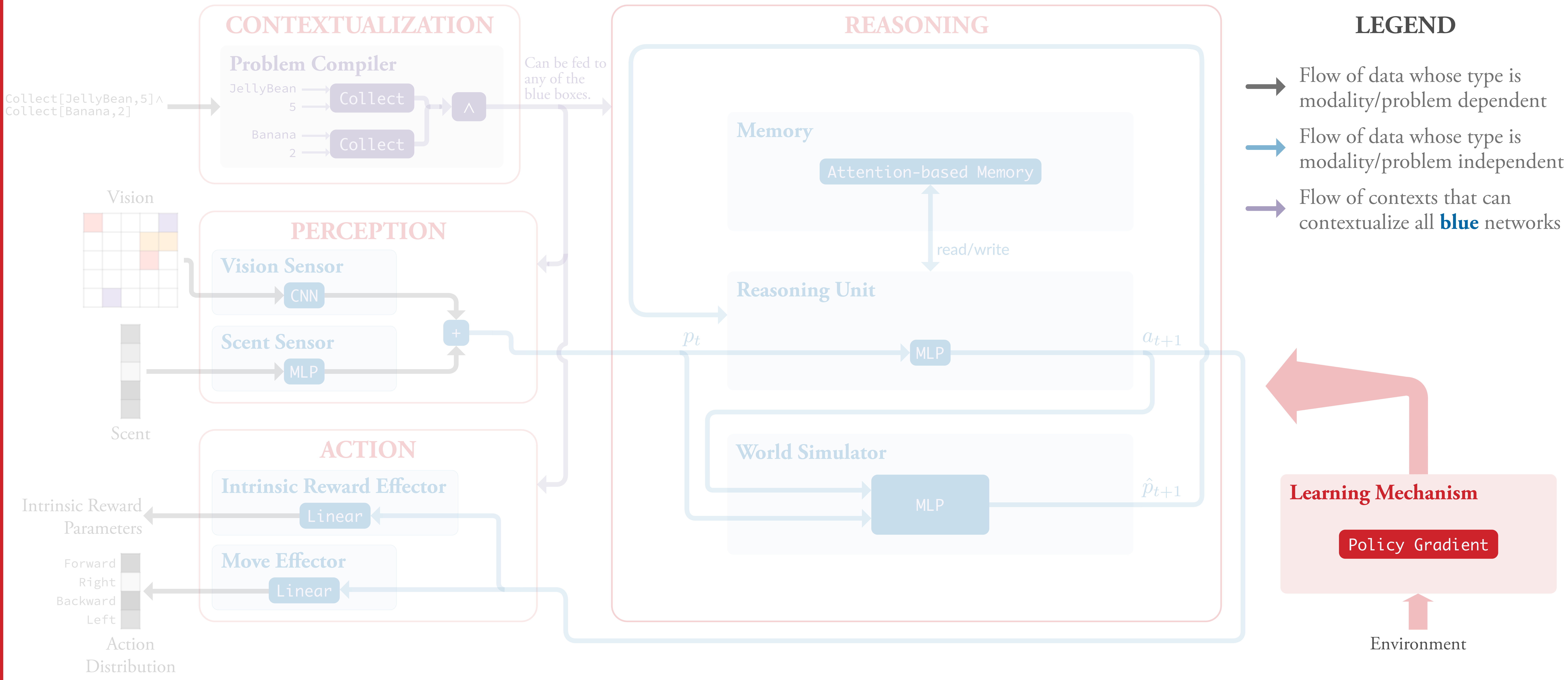
$$[a_{t+1}, s_{t+1}, \mathrm{STOP}_{t+1}] = R(p_t, a_t, s_t)$$



LEGEND

- Flow of data whose type is modality/problem dependent
- Flow of data whose type is modality/problem independent
- Flow of contexts that can contextualize all **blue** networks

Unified Architecture: JBW Example



Our approach relies on *gradient-based* updates:

- Actions can have *feedback mechanisms:*

$$h(f_\theta(x), \mathcal{E})$$

- Whenever a feedback mechanism triggers, a gradient-based update is performed along:

$$\text{direction} \triangleq h(f_\theta(x), \mathcal{E}) \frac{\partial f_\theta(x)}{\partial \theta}$$

EXTERNAL

INTERNAL

SUPERVISED LEARNING

$$\frac{\partial \mathcal{L}(f_\theta(x), \mathcal{E})}{\partial f_\theta(x)}$$

REINFORCEMENT LEARNING

Advantage Function

We also plan to explore *staged learning* and *curriculum learning.*

[**Platanios**, Stretcu, Neubig, Poczos, Mitchell, NAACL 2019]

Goal/Problem Specification

Input

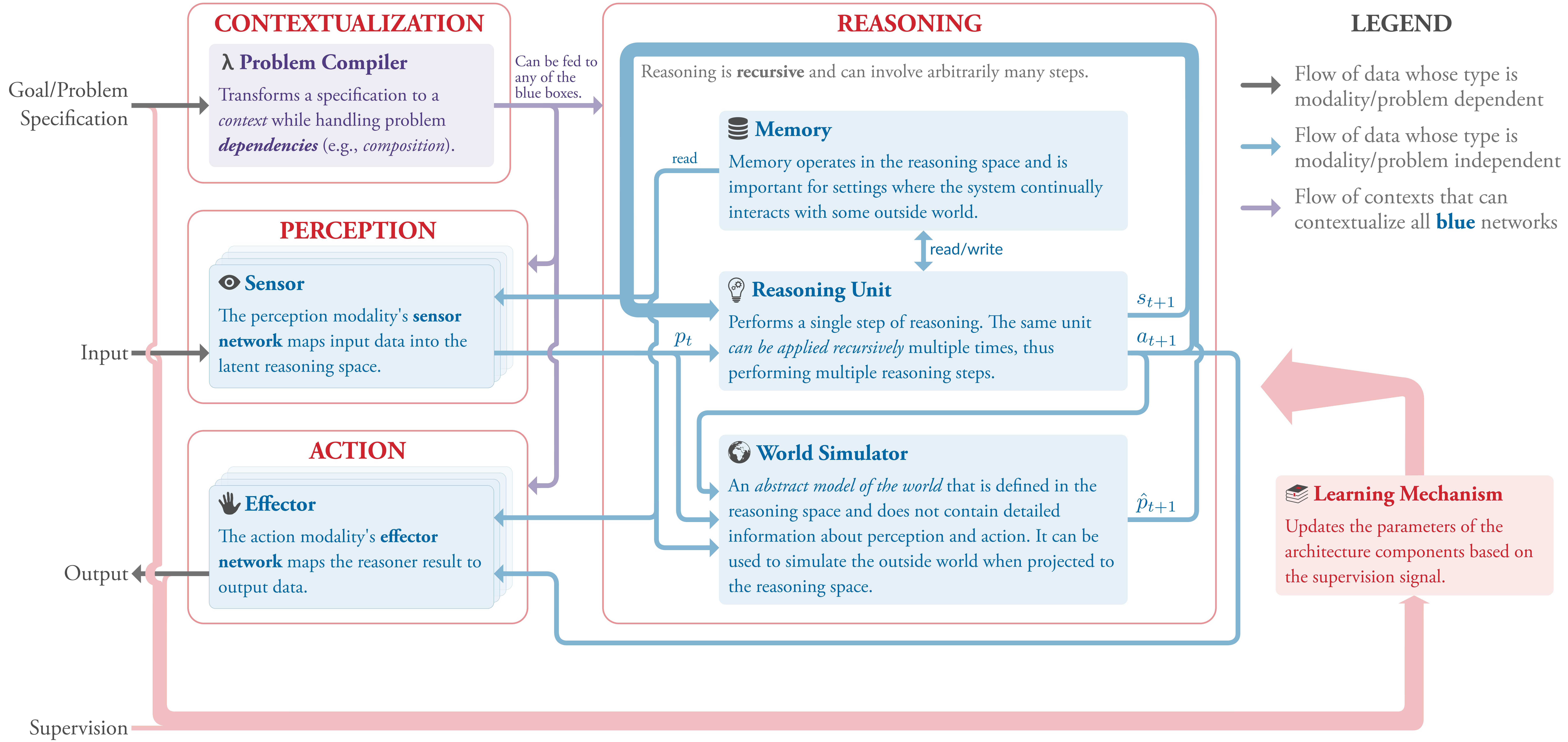
Output

Supervision

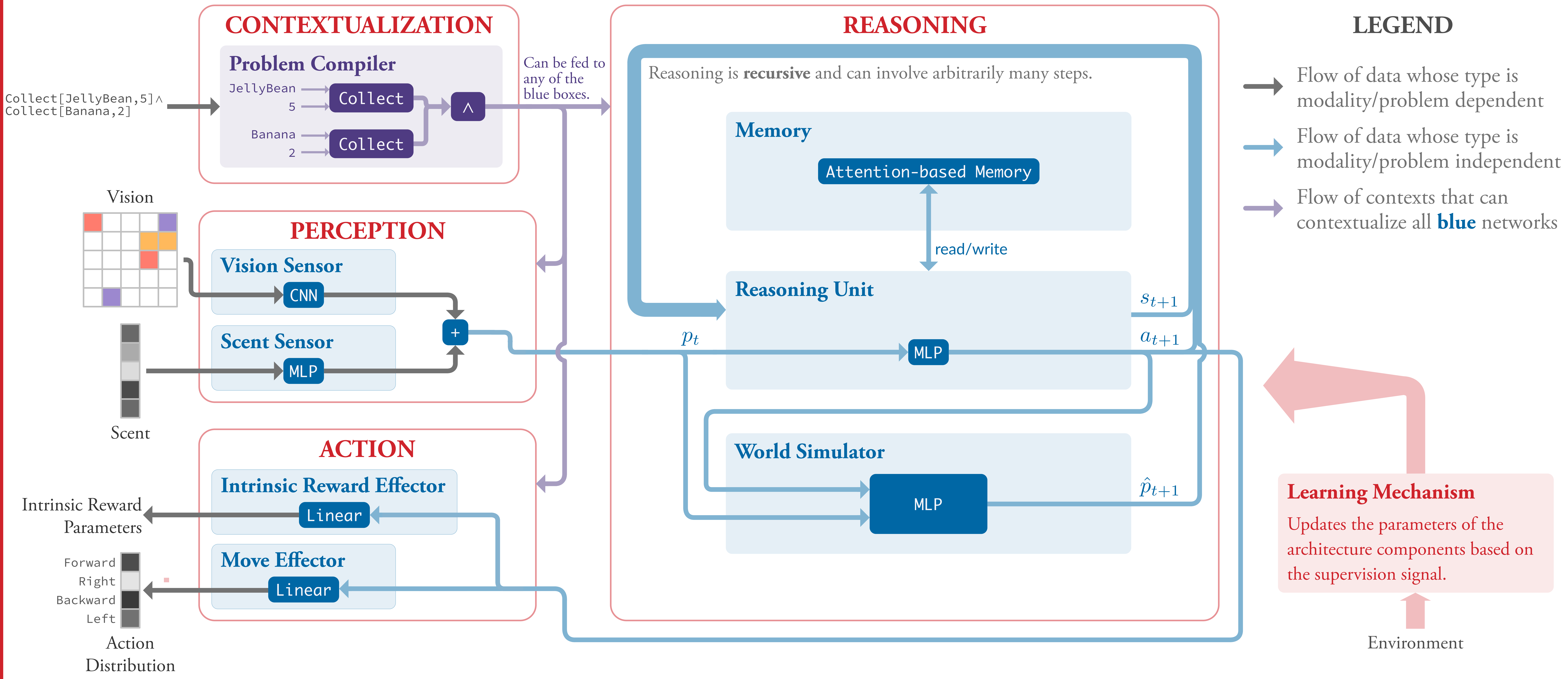
Learning Mechanism

Updates the parameters of the architecture components based on the supervision signal.

Unified Architecture



Unified Architecture: JBW Example



Outline

Part 1: Proposed Approach

1. Learning from Noisy Labels
   supervision noise and ambiguity
2. Contextual Parameter Generation
   multi-task learning
3. Self-Reflection
   unsupervised learning
4. Unified Architecture
   general learning and intelligence

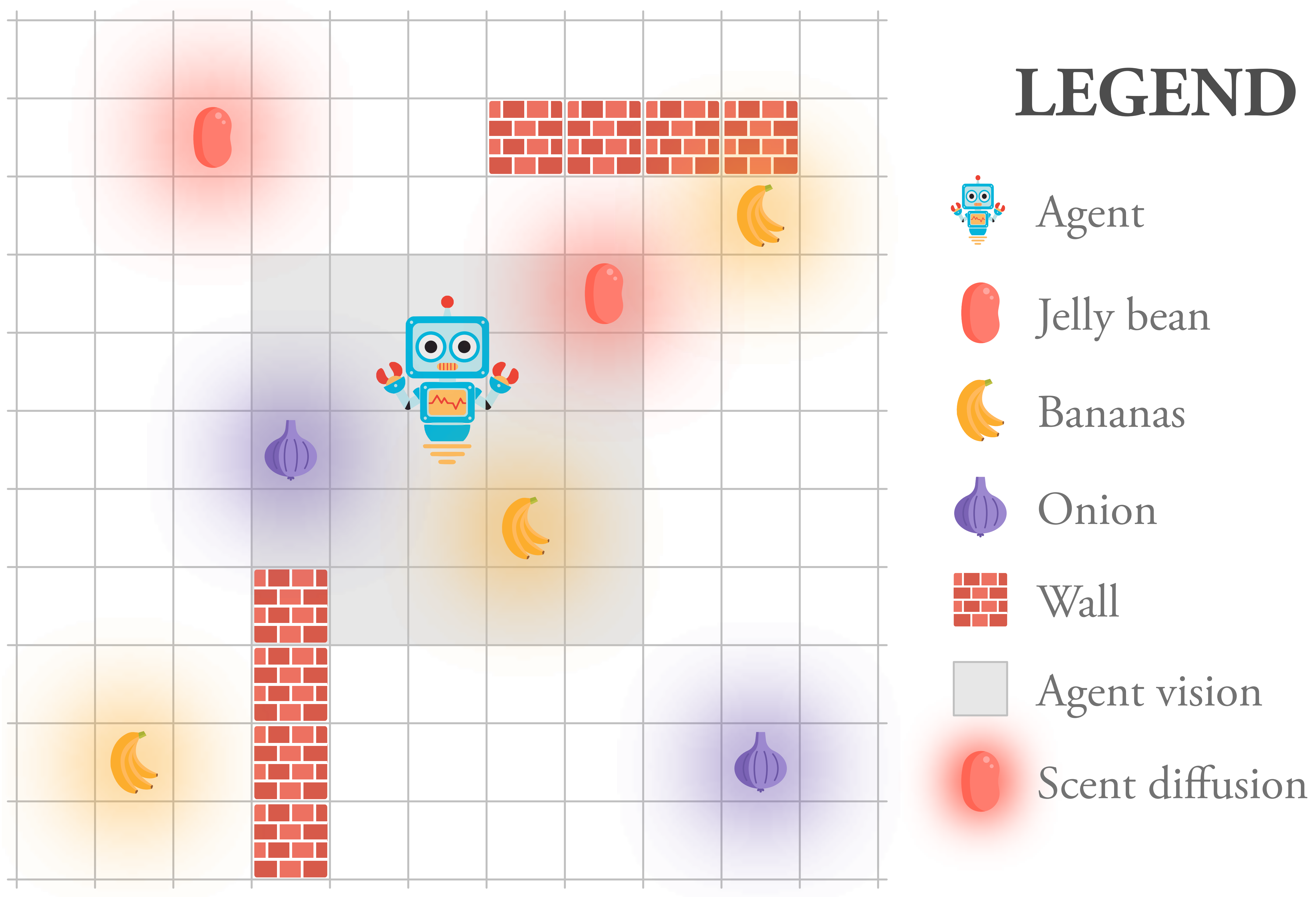
Part 2: Evaluation

Part 3: Timeline

Evaluation

We designed a world to help us test our thesis statement:

- Procedurally Generated
- Multi-Modal:
  - **Vision:** high precision / low recall
  - **Scent:** low precision / high recall
- Multi-Task:
  - Provided
  - Compositional
- Continual Learning
- Ever-Changing Learning Problems



Proposed Case Studies

We propose to perform the following case studies:

- JBW Provided Problems: Tests mixed learning paradigm.
  - Positive reward for some items and negative for others.
  - E.g., Collect[JellyBean] $\wedge$ $\neg$Collect[Onion].
  - Auxiliary item recognition tasks from color/scent.
- JBW Generated Problems: Tests problem generation and goal contextualization.
- JBW Advanced Problems: Tests memory and recursive reasoning.
- Atari: Tests structured information sharing across multiple tasks.
- NLP: Tests large scale multi-modal and multi-task learning ability.

Outline

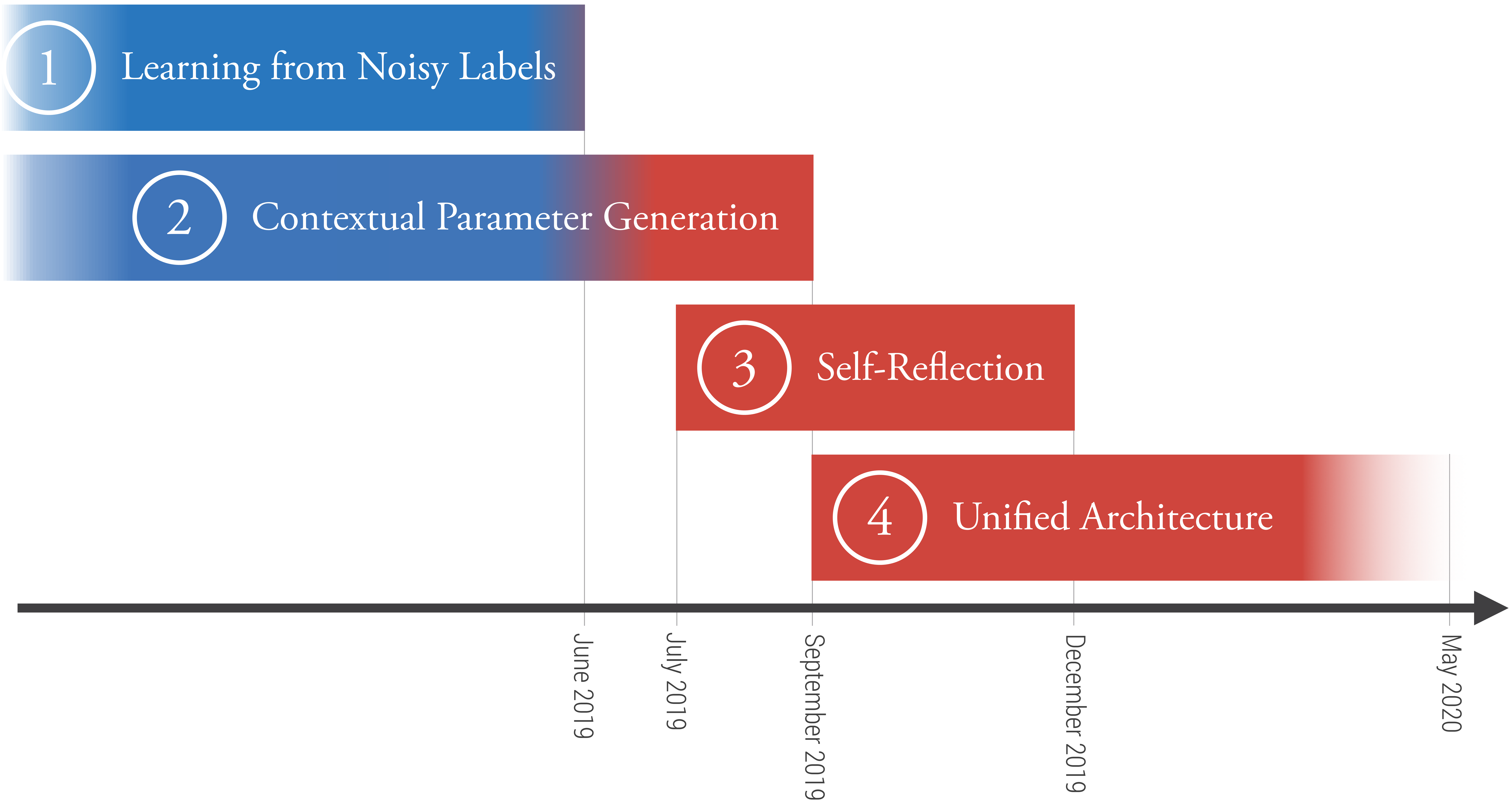
Part 1: Proposed Approach

1. Learning from Noisy Labels
   supervision noise and ambiguity
2. Contextual Parameter Generation
   multi-task learning
3. Self-Reflection
   unsupervised learning
4. Unified Architecture
   general learning and intelligence

Part 2: Evaluation

Part 3: Timeline

Timeline



1 Learning from Noisy Labels
2 Contextual Parameter Generation
3 Self-Reflection
4 Unified Architecture
June 2019
July 2019
September 2019
December 2019
May 2020

Publications

Learning from Noisy Labels

(1) *Estimating Accuracy from Unlabeled Data.*
Emmanouil A. Platanios, A. Blum, and T. Mitchell.
In Uncertainty in Artificial Intelligence (**UAI**), 2014.

(2) *Estimating Accuracy from Unlabeled Data: A Bayesian Approach.*
Emmanouil A. Platanios, A. Dubey, and T. Mitchell.
In International Conference in Machine Learning (**ICML**), 2016.

(3) *Estimating Accuracy from Unlabeled Data: A Probabilistic Logic Approach.*
Emmanouil A. Platanios, H. Poon, T. Mitchell, and E. Horvitz.
In Neural Information Processing Systems (**NeurIPS**), 2017.

(4) *Learning from Multiple Noisy Labels.*
Emmanouil A. Platanios, M. Al-Shedivat, E. Xing, and T. Mitchell.
Under review for Neural Information Processing Systems (**NeurIPS**), 2019.

Contextual Parameter Generation

(5) *Contextual Parameter Generation for Universal Neural Machine Translation.*
Emmanouil A. Platanios, M. Sachan, G. Neubig, and T. Mitchell.
In Empirical Methods in Natural Language Processing (**EMNLP**), 2018.

(6) *Contextual Parameter Generation for Knowledge Graph Link Prediction.*
Emmanouil A. Platanios*, O. Stretcu*, G. Stoica*, B. Poczos, and T. Mitchell.
Under review for Empirical Methods in Natural Language Processing (**EMNLP**), 2019.

Curriculum Learning

(7) *Competence-based Curriculum Learning for Neural Machine Translation.*
Emmanouil A. Platanios, O. Stretcu, G. Neubig, B. Poczos, and T. Mitchell.
In the Conference of the North American Chapter of the Association for Computational Linguistics (**NAACL**), 2019.

Open-Source Projects

TensorFlow Scala

Watch 62 | Star 661 | Fork 65

https://github.com/eaplatanios/tensorflow_scala

Symphony MT

Watch 7 | Star 24 | Fork 2

https://github.com/eaplatanios/symphony-mt

Never-Ending Learning

(8) *Never-Ending Learning.*
T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohammad, N. Nakashole, **Emmanouil A. Platanios**, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and M. Welling.
In Communications of the ACM (**CACM**), 2018.

(9) *Never-Ending Learning.*
T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohammad, N. Nakashole, **Emmanouil A. Platanios**, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and M. Welling.
In AAAI Conference on Artificial Intelligence (**AAAI**), 2015.

Other

(10) *Graph Agreement Models for Semi-Supervised Learning.*
O. Stretcu, K. Viswanathan, D. Movshovitz-Attias, **Emmanouil A. Platanios**, S. Ravi, and A. Tomkins.
Under review for Neural Information Processing Systems (**NeurIPS**), 2019.