# Provable Representation Learning

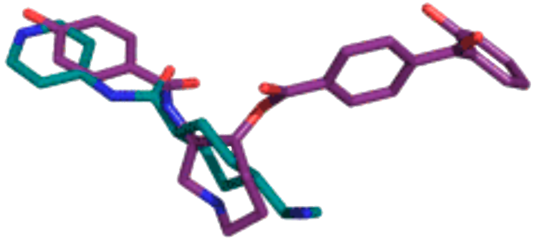
**Simon S. Du**

07-28-2021

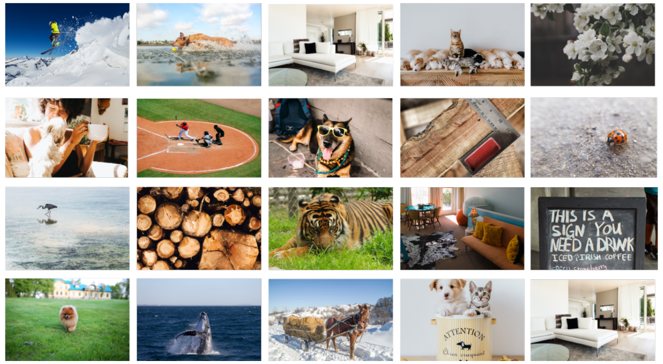
IST Seminar Series Mathematics, Physics & Machine Learning

# Neural Networks

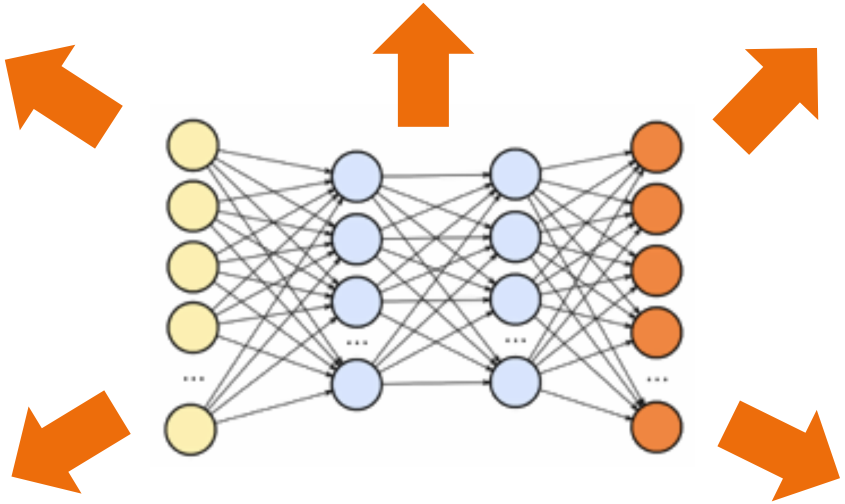

Natural Language Processing
(Recurrent NN, Attention NN)



Bioinformatics
(Graph NN)



Computer Vision
(Convolutional NN)



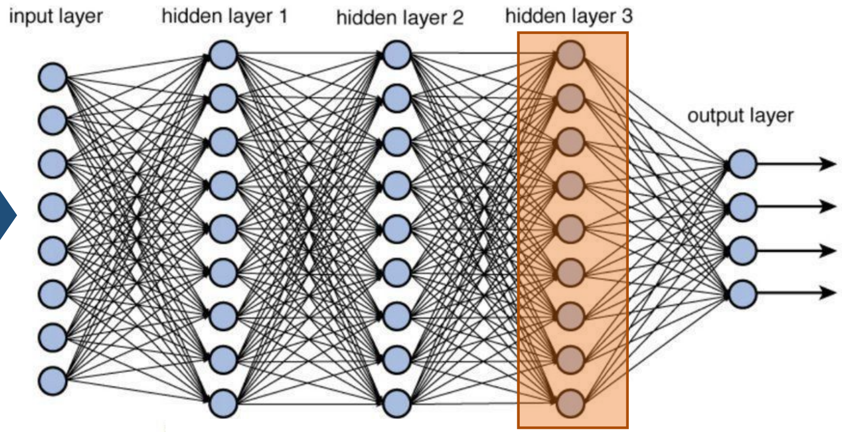
| Size (feet$^2$) $x_1$ | # of rooms $x_2$ | Age (years) $x_3$ | Price ($1000) $y$ |
|---|---|---|---|
| 2104 | 5 | 45 | 460 |
| 1416 | 3 | 40 | 232 |
| 1543 | 3 | 36 | 315 |

Feature-based Data
(Fully Connected NN)


Reinforcement Learning
(Policy NN, Q NN)

# Representation Learning in CV

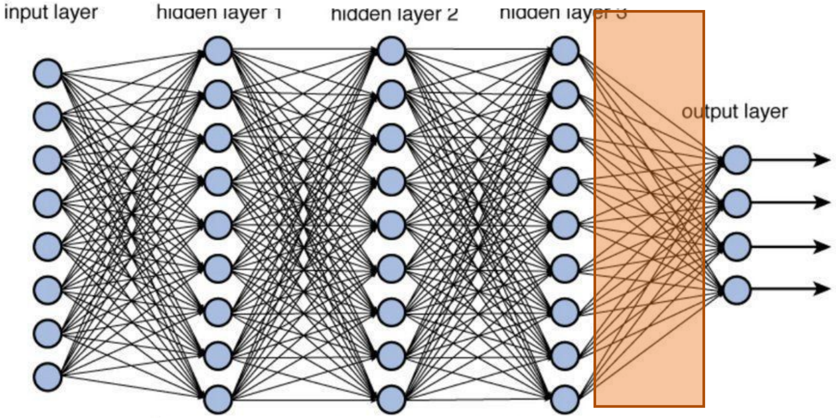
Image Representation




Cat

Train a neural network (ResNet) on ImageNet (1M data, 1000 classes)

**Representation (feature extractor):** The mapping from image to the second-to-the-last layer.



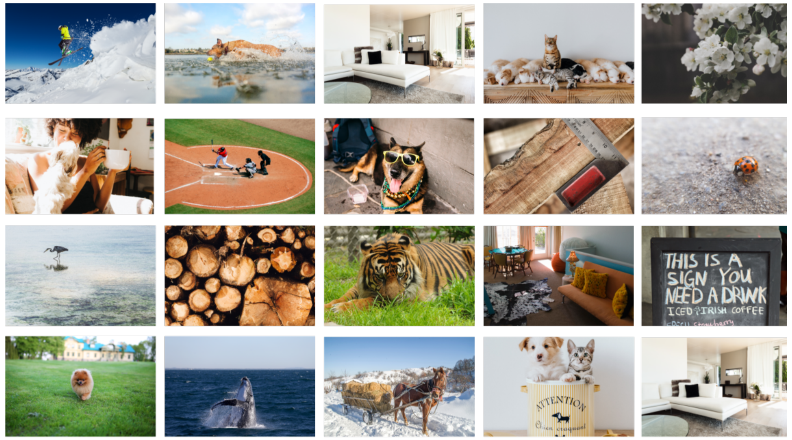

Dog

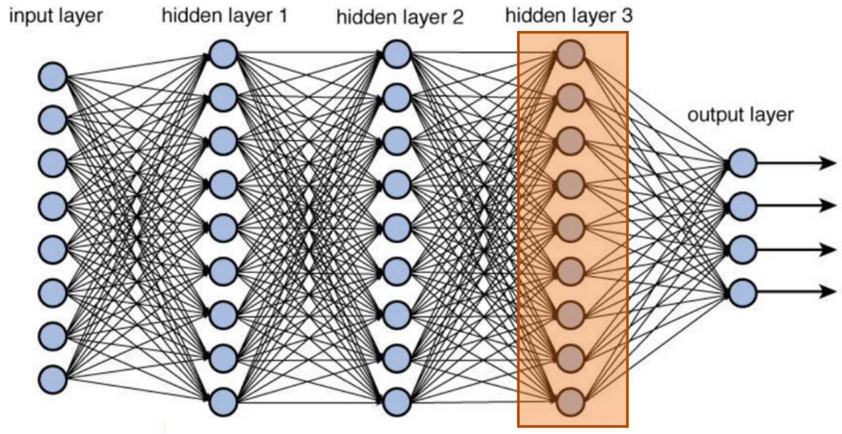
Fix the representation, just re-train the last linear layer.

New linear classifier

# Example

**Source tasks**
(for training representation):
ImageNet




ResNet

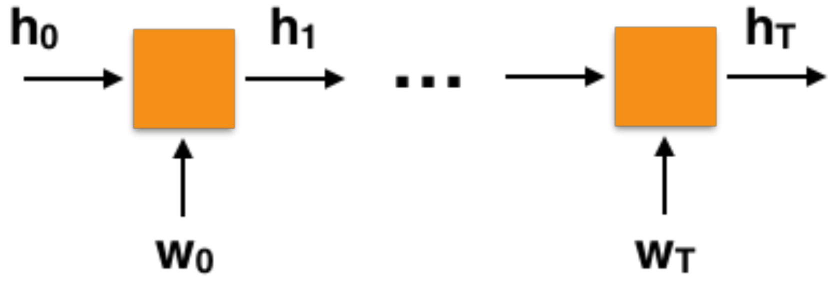
**Target task:**
Few-shot Learning on VOC07 dataset (20 classes, 1-8 examples per class)


- Without representation learning: **5% - 10%** (random guess = **5%)**
- With representation learning: **50% - 80%**

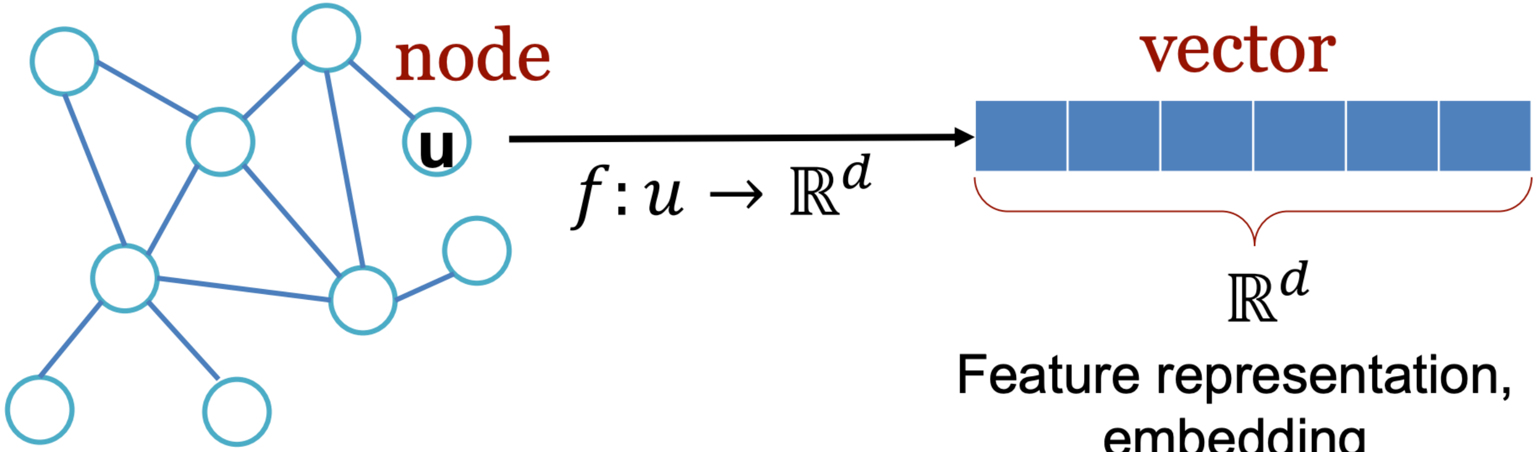
# Examples

Natural Language Processing


Final hidden state: Sentence representation




Graph Representation Learning




Feature representation, embedding

[Bengio et al. 2012]

# Two Questions

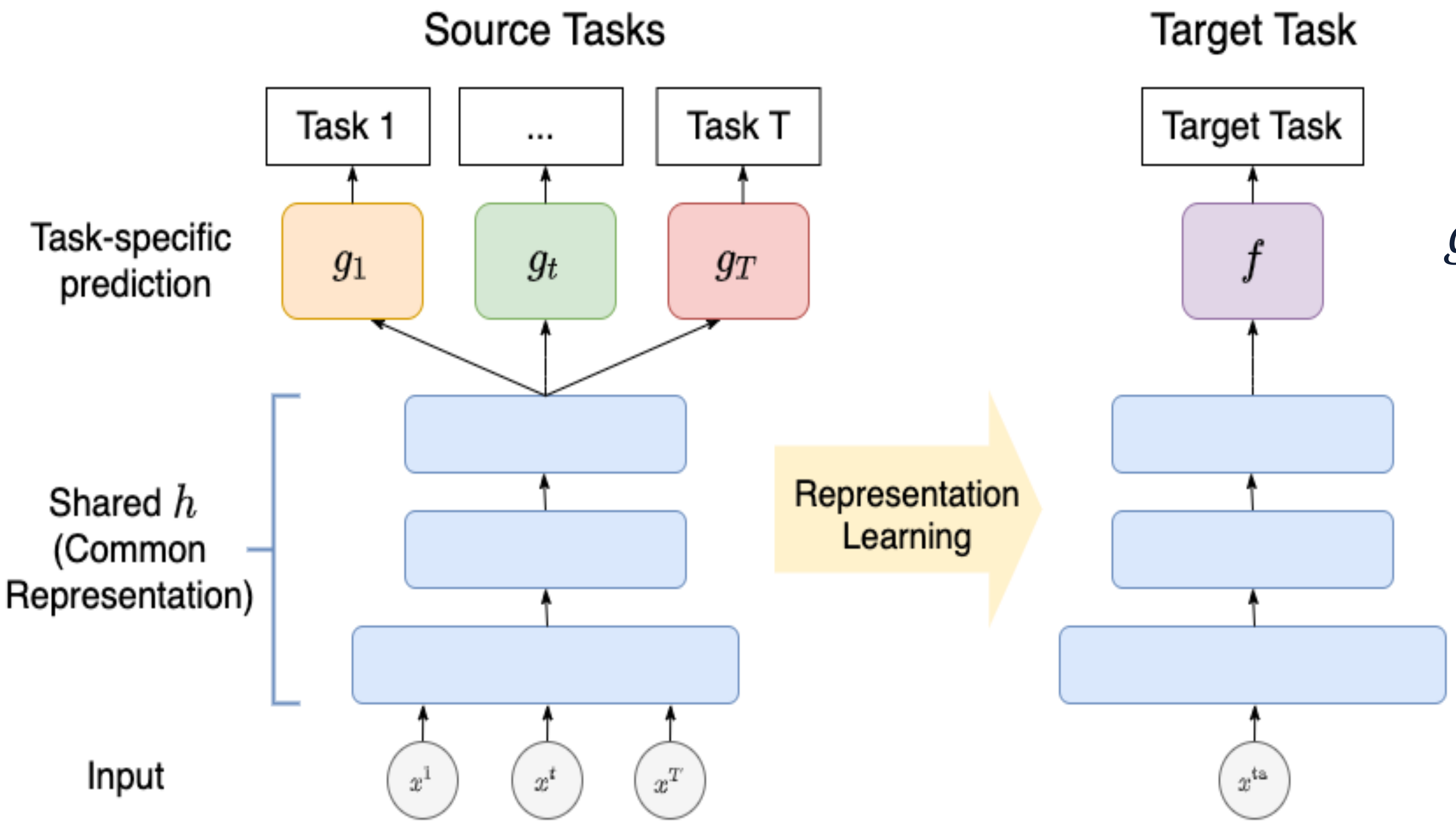
## Q1: When?

What are the necessary and sufficient conditions?

## Q2: Why?

What is the mechanism?

# Formulation




$g_1, \dots g_T, f \in \mathcal{G}$: prediction class (e.g., linear classifier)

$h \in \mathcal{H}$: representation class (e.g., multi-layer NN)

# Formulation

## Representation Learning

- $T$ source tasks, each with $\boldsymbol{n_1}$ data:
  $\left\{(x_1^t, y_1^t) \dots \left(x_{n_1}^t, y_{n_1}^t\right)\right\}_{t=1}^T$

- Learning representation:
  $$\min_{h \in \mathcal{H}} \sum_{t=1}^{T} \min_{g_t \in \mathcal{G}} \sum_{i=1}^{n_1} \ell\left(g_t\left(h\left(x_i^t\right)\right), y_i^t\right)$$

$\ell$: quadratic loss

## Predictor Learning

- 1 target task, with $\boldsymbol{n_2} \ll \boldsymbol{n_1}$ data:
  $\left(x_1^{ta}, y_1^{ta}\right) \dots \left(x_{n_2}^{ta}, y_{n_2}^{ta}\right) \sim \mu$

- Training for the target task:
  $$\min_{f \in \mathcal{G}} \sum_{i=1}^{n_2} \ell\left(f\left(h\left(x_i^t\right)\right), y_i^t\right)$$

Representation $\boldsymbol{h}(\cdot)$ is fixed

# Standard Statistical Learning Theory

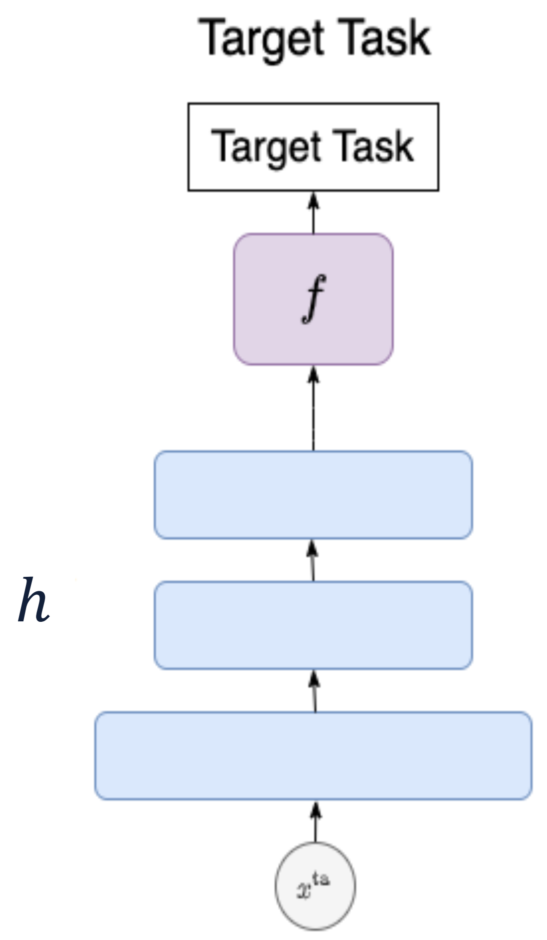
Training with data only from the target domain:

$$\min_{f \in \mathcal{G}, h \in \mathcal{H}} \sum_{i=1}^{n_2} \ell\left(f\left(h\left(x_i^{ta}\right)\right), y_i^{ta}\right)$$

**Theorem （ Example ）**

$$\mathbb{E}_{(x^{ta}, y^{ta}) \sim \mu}\left[\ell\left(f\left(h\left(x^{ta}\right)\right), y^{ta}\right)\right] = O\left(\frac{\mathcal{C}(\mathcal{H}) + \mathcal{C}(\mathcal{G})}{n_2}\right)$$

$\mathcal{C}(\mathcal{H})$: complexity measure of the representation class.
$\mathcal{C}(\mathcal{G})$: complexity measure of the prediction class.
E.g., # of variables (linear function class), VC-dimension, Rademacher complexity, Gaussian width, etc



# Ideal Theory for Representation Learning

Identify a set of (natural) assumptions:

1. If the data satisfies these assumptions, representation learning provably helps.
2. Without assumptions, representation learning does not help.

**Theorem (Example)**

$$\mathbb{E}_{(x^{ta},y^{ta})\sim\mu}\left[\ell\left(f\left(h(x^{ta})\right),y^{ta}\right)\right] = O\left(\frac{\mathcal{C}(\mathcal{H})}{\boldsymbol{n_1 T}} + \frac{\mathcal{C}(\mathcal{G})}{n_2}\right)$$

When **# of tasks ($\boldsymbol{T}$)** is larger, much better than

$$O\left(\frac{\mathcal{C}(\mathcal{H})+\mathcal{C}(\mathcal{G})}{n_2}\right)$$

for learning the representation

for learning the predictor

# Asmp 1: Existence of a Good Representation

**Assumption 1: Existence of a Good Representation**

There exist a representation $h^* \in \mathcal{H}$ and predictors $g_1^*, g_2^*, \dots, g_T^*, f^* \in \mathcal{G}$ such that

$$\mathbb{E}_{(x_t, y_t)\sim\mu_t}\left[\ell\left(g_t^*\left(\boldsymbol{h}^*(x_t)\right), y_t\right)\right] = 0 \ \forall t = 1, \dots, T$$

$$\mathbb{E}_{(x_{ta}, y_{ta})\sim\mu}\left[\ell\left(f^*\left(\boldsymbol{h}^*(x_{ta})\right), y_{ta}\right)\right] = 0$$

A **shared** good representation for all source tasks and the target task:

This is why we use representation learning.

(Without this assumption, we should not use representation learning)

# Existence of Good Rep is NOT Enough

**Source tasks:**
**Classify types of**
**cats.**

**Target task:**
**Cat or dog?**

Source tasks can learn a good representation for cats,
but not a good representation for **both cats and dogs**.

# Existence of Good Rep is NOT Enough

Input: 1000 dimensional 0/1 vector, $\{0,1\}^{1000}$

Good representation: first 100 dimension

- All tasks (source and target) only need first 100 digits for accurate prediction.
- Predicting whether the $10^{th}$-digit is 1, predicting the sum of first 100 digits, etc.

**Bad scenario:**

- Source tasks only need to use first 50 digits: e.g., whether the $10^{th}$-digit is 1
- Target tasks need to use **all** first 100 digits: e.g., predicts the sum of first 100 digits

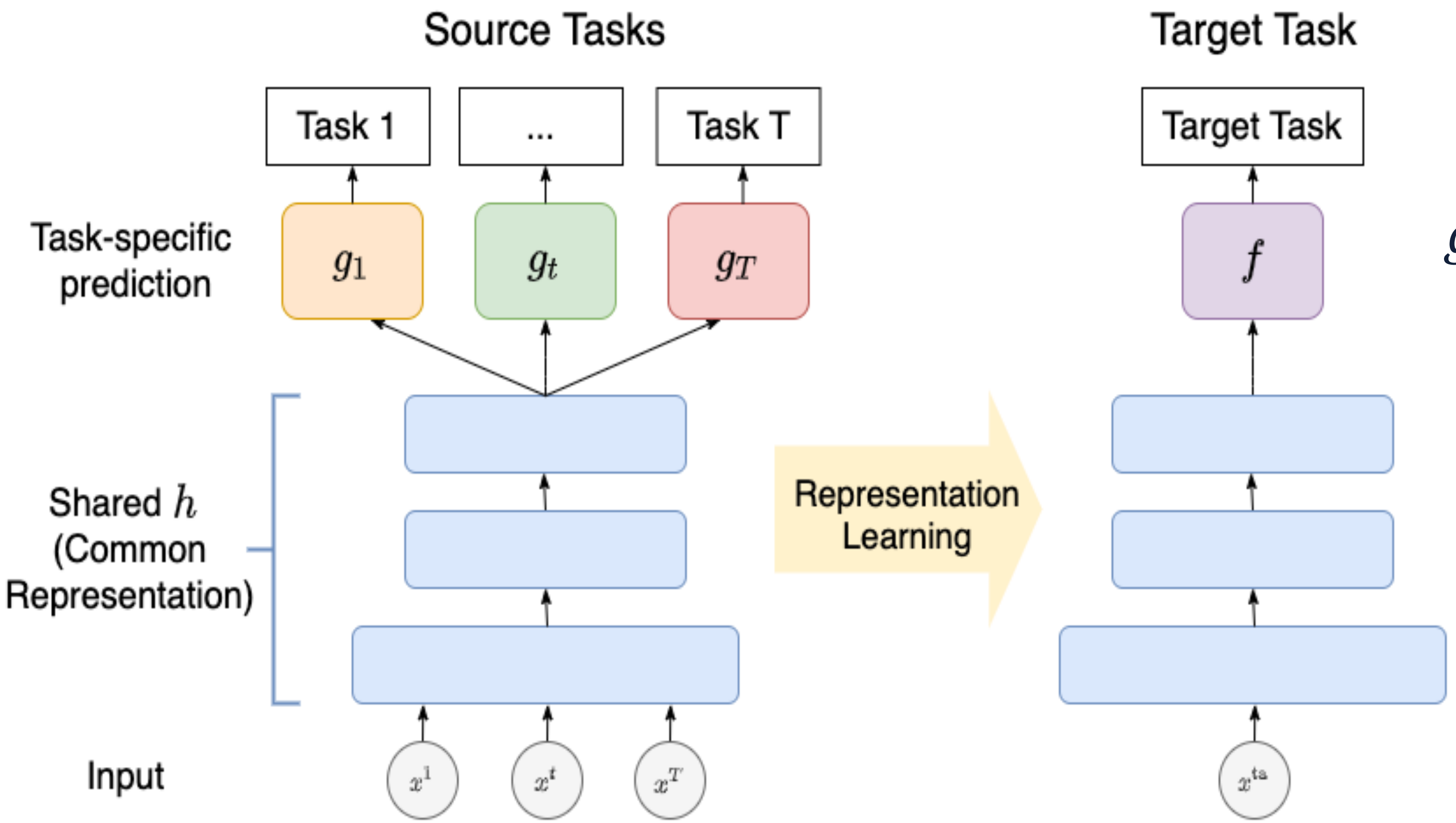
Source tasks cannot give the **full information** about the good representation!

# Assumption 2: Diversity of Source Tasks

Representation learning is useful only if source tasks can give the full information about the good representation, a.k.a., **diversity of the source tasks.**

**Q:** **What is the definition of diversity?**

# Formulation




$g_1, \dots g_T, f \in \mathcal{G}$: prediction class (e.g., linear classifier)

$h \in \mathcal{H}$: representation class (e.g., multi-layer NN)

# Diversity for Linear Predictors

$\mathcal{G}$: linear prediction class (last layer of neural networks)

**Assumption 1: Existence of a Good Representation**

There exist a representation $h^* \in \mathcal{H}, h^*(x) \in \mathbb{R}^k$ and $w_1^*, w_2^*, \dots, w_T^*, w_{ta}^* \in \mathbb{R}^k$:

$$\mathbb{E}_{(x_t,y_t)\sim\mu_t}[\ell(\langle w_t^*, h^*(x_t)\rangle, y_t)] = 0 \ \forall t = 1, \dots, T$$

$$\mathbb{E}_{(x_{ta},y_{ta})\sim\mu}[\ell(\langle w_{ta}^*, h^*(x_{ta})\rangle, y_{ta})] = 0$$

**Assumption 2: Diversity of Source Tasks for Linear Predictor**

$W^* = [w_1^*, w_2^*, \dots, w_T^*] \in \mathbb{R}^{k\times T}$ is full rank (=k).

Need $T \geq k$: cover the **span** of the good representation.

# Linear Representation (Subspace Learning)

Input: $x \in \mathbb{R}^d$. Linear representation class $\mathcal{H}$: matrices of size $k \times d$ $(k \ll d)$.

## Assumption 1: Existence of a Good Representation

There exists a linear representation $B^* \in \mathbb{R}^{k \times d}$, and $w_1^*, w_2^*, \ldots, w_T^*, w_{ta}^* \in \mathbb{R}^k$:

$$\mathbb{E}_{(x_t, y_t) \sim \mu_t}[\ell(\langle w_t^*, B^* x_t \rangle, y_t)] = 0 \ \forall t = 1, \ldots, T$$

$$\mathbb{E}_{(x_{ta}, y_{ta}) \sim \mu}[\ell(\langle w_{ta}^*, B^* x_{ta} \rangle, y_{ta})] = 0$$

## Theorem [D. Hu Kakade Lee Lei, 2020]

Under Assumption 1 &2, we have $\mathbb{E}_{(x^{ta}, y^{ta}) \sim \mu}\left[\ell\left(f\left(h(x^{ta})\right), y^{ta}\right)\right] = O\left(\frac{dk}{\boldsymbol{n_1 T}} + \frac{k}{n_2}\right)$.

Without representation learning, directly learning a linear predictor on $\mathbb{R}^d$: $O\left(\frac{d}{n_2}\right)$.

# Main Result for General Representation Class

## Assumption 1: Existence of a Good Representation

There exist a representation $h^* \in \mathcal{H}, h^*(x) \in \mathbb{R}^k$ and $w_1^*, w_2^*, \ldots, w_T^*, w_{ta}^* \in \mathbb{R}^k$:

$$\mathbb{E}_{(x_t, y_t) \sim \mu_t}[\ell(\langle w_t^*, h^*(x_t) \rangle, y_t)] = 0 \ \forall t = 1, \ldots, T$$

$$\mathbb{E}_{(x_{ta}, y_{ta}) \sim \mu}[\ell(\langle w_{ta}^*, h^*(x_{ta}) \rangle, y_{ta})] = 0$$

## Theorem [D. Hu Kakade Lee Lei, 2020]

Under Assumption 1 &2, we have $\mathbb{E}_{(x^{ta}, y^{ta}) \sim \mu}\left[\ell\left(f\left(h(x^{ta})\right), y^{ta}\right)\right] = O\left(\frac{\mathcal{C}(\mathcal{H})}{\boldsymbol{n_1 T}} + \frac{k}{n_2}\right)$.

$\mathcal{C}(\mathcal{H})$: Gaussian width of the representation class $\mathcal{H}$.

- Measures how well the function in the class can fit the noise.

# Comparison with Previous Work

**Theorem [D. Hu Kakade Lee Lei 2020]**

Under Assumption 1 &2, we have $\mathbb{E}_{(x^{ta},y^{ta})\sim\mu}\left[\ell\left(f\left(h(x^{ta})\right),y^{ta}\right)\right] = O(\frac{dk}{\boldsymbol{n_1 T}} + \frac{k}{n_2})$.

**Theorem [Maurer Pontil Romera-Paredes 2016]**

Under Assumption 1, and that **all tasks (source and target) are i.i.d. sampled** from a distribution over tasks,

we have $\mathbb{E}_{(x^{ta},y^{ta})\sim\mu}\left[\ell\left(f\left(h(x^{ta})\right),y^{ta}\right)\right] = O(\frac{dk}{\boldsymbol{T}} + \frac{k}{n_2})$.

$O\left(\frac{1}{T}\right)$, instead of $O\left(\frac{1}{n_1 T}\right)$, is tight for the setting in [Mauer et al. 2016].

# Why Does Rep learning Help: Proof Intuition

Joint optimization of representation and prediction:

$$\min_{h\in\mathcal{H}} \sum_{t=1}^{T} \min_{g_t\in\mathcal{G}} \sum_{i=1}^{n_1} \ell\left(g_t\left(h\left(x_i^t\right)\right), y_i^t\right)$$

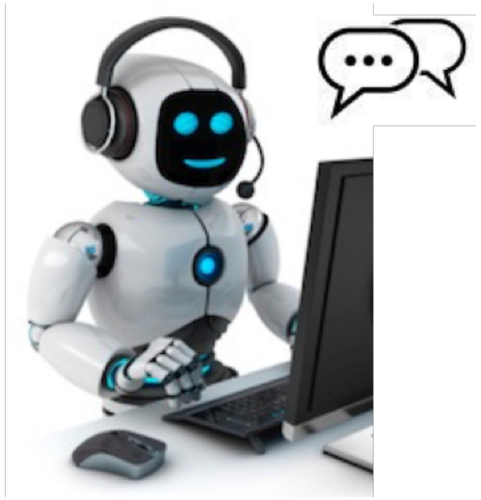
**Main Ideas:**

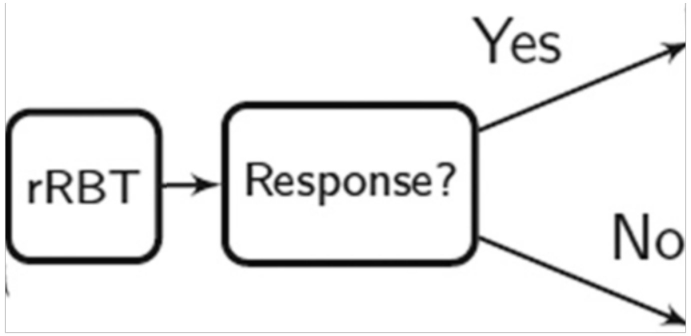
- Optimization on representation is over **all** tasks.
- We must find a **shared** good representation for all tasks, otherwise, the loss cannot be small: joint optimization forces to learn a good representation.

# Key Message

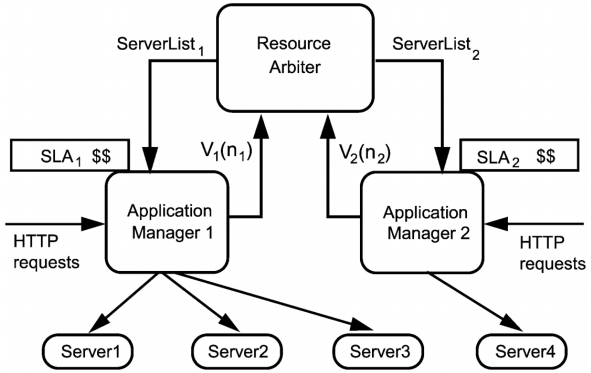
**Existence of a good representation** and **diversity of tasks** are key conditions that enable **representation learning** to improve sample efficiency.

# Reinforcement Learning









[Levine et al 16] [Ng et al 03]

[Mandel et al 14]

[Singh et al 02]

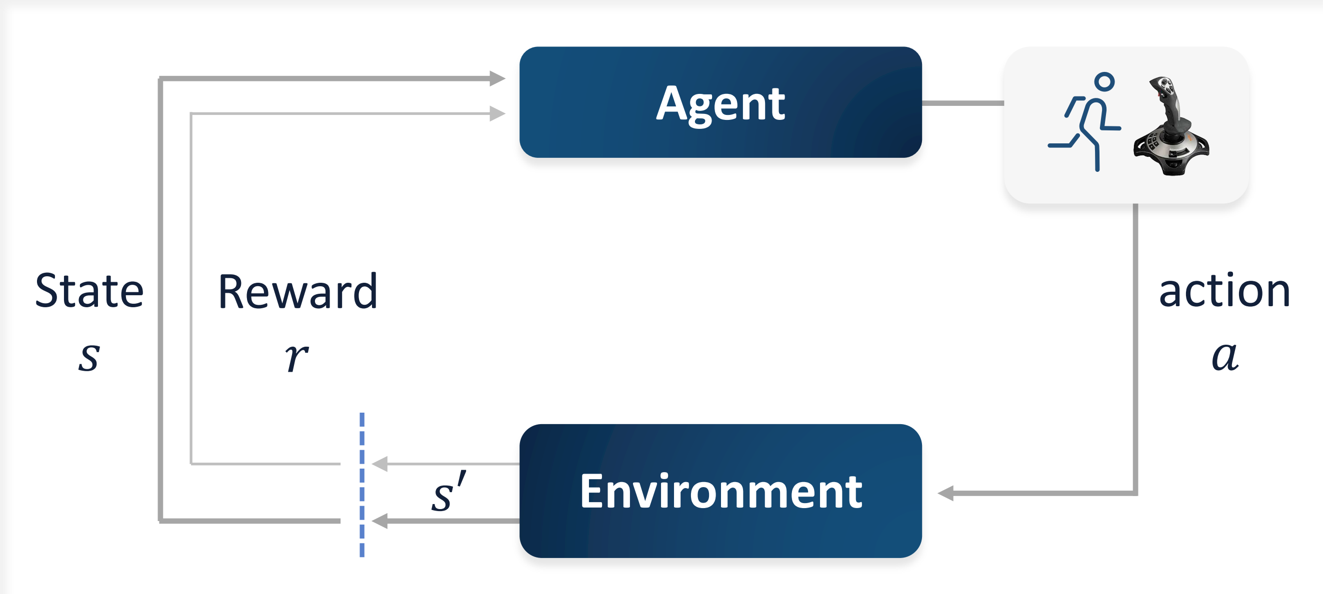
[Tesauro et al 07]

[Let et al 12]

[Minh et al 15] [Silver et al 16]

# Reinforcement Learning

## Markov Decision Process




A policy $\pi$ : $\mathrm{States(S)} \rightarrow \mathrm{Actions\ (A), a} = \pi(s)$

Goal: maximize the expected total reward $\mathbb{E}\left[r_1 + r_2 + \cdots \mid \pi\right]$

$\pi^*$: optimal policy (maximizes the expected total reward)

# Multi-task Reinforcement Learning




Autonomous driving on different situations

A class of different but **related** tasks.

- Each task has a different optimal policy.
- Share the same state space and action space.

# Imitation Learning

Trajectories from the optimal policy $\pi^*$ (expert) are available:

$$\{(s_i, \pi^*(s_i))\}_{i=1}^{n}$$

Multi-task Imitation learning:

T source tasks, each task we have $n_1$ samples from experts.

1 target task with $n_2$ samples from the expert.

# Representation Learning for Imitation Learning

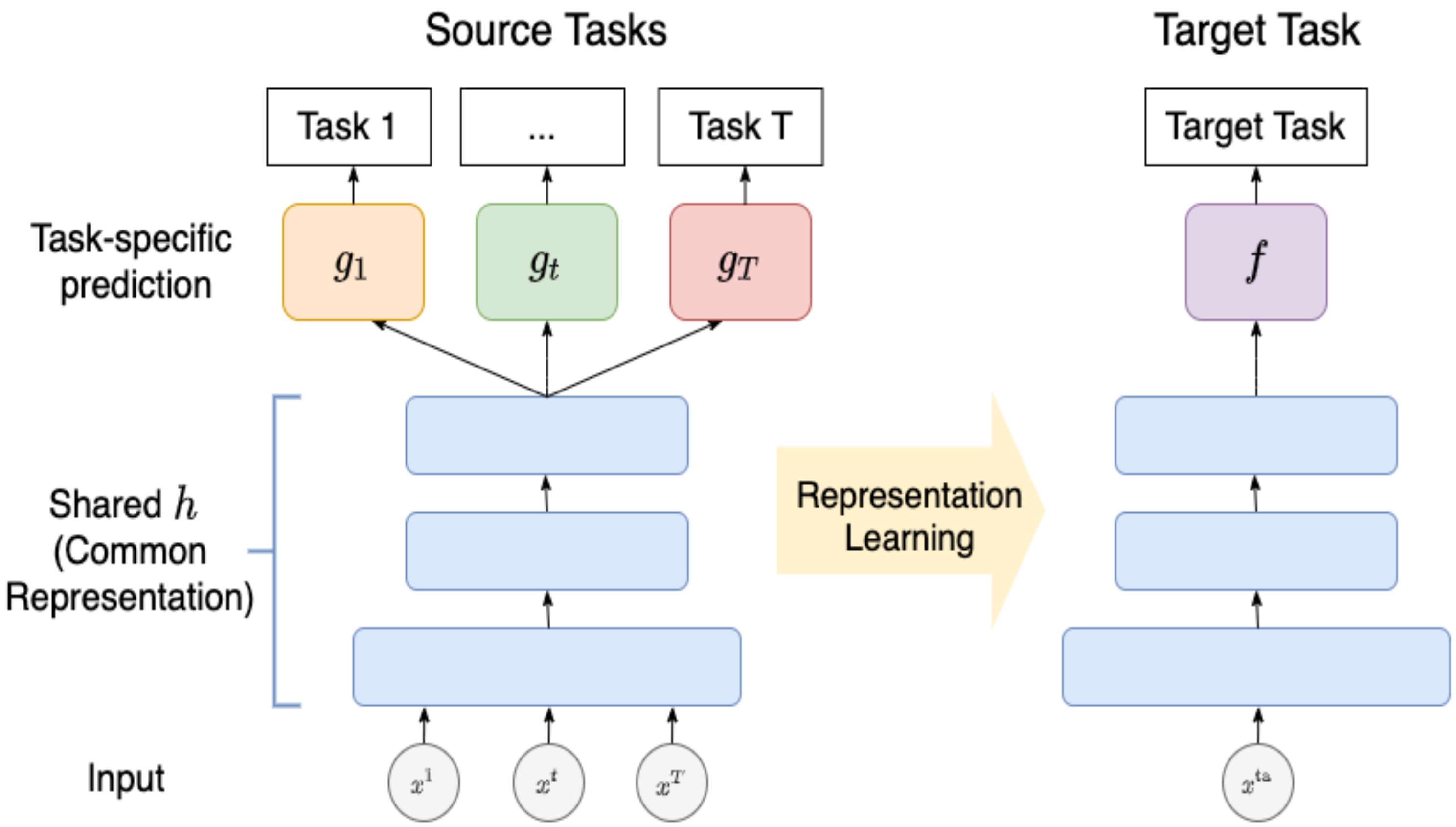
Model the optimal policy $\pi^*$:

- Source tasks,t $= 1, \ldots, \mathrm{T}$:

$$\pi_t^*(s) = g_t\big(h(s)\big)$$

- Target task:

$$\pi_{ta}^*(s) = f\big(h(s)\big)$$



# Representation Learning for Imitation Learning

Learning representation:

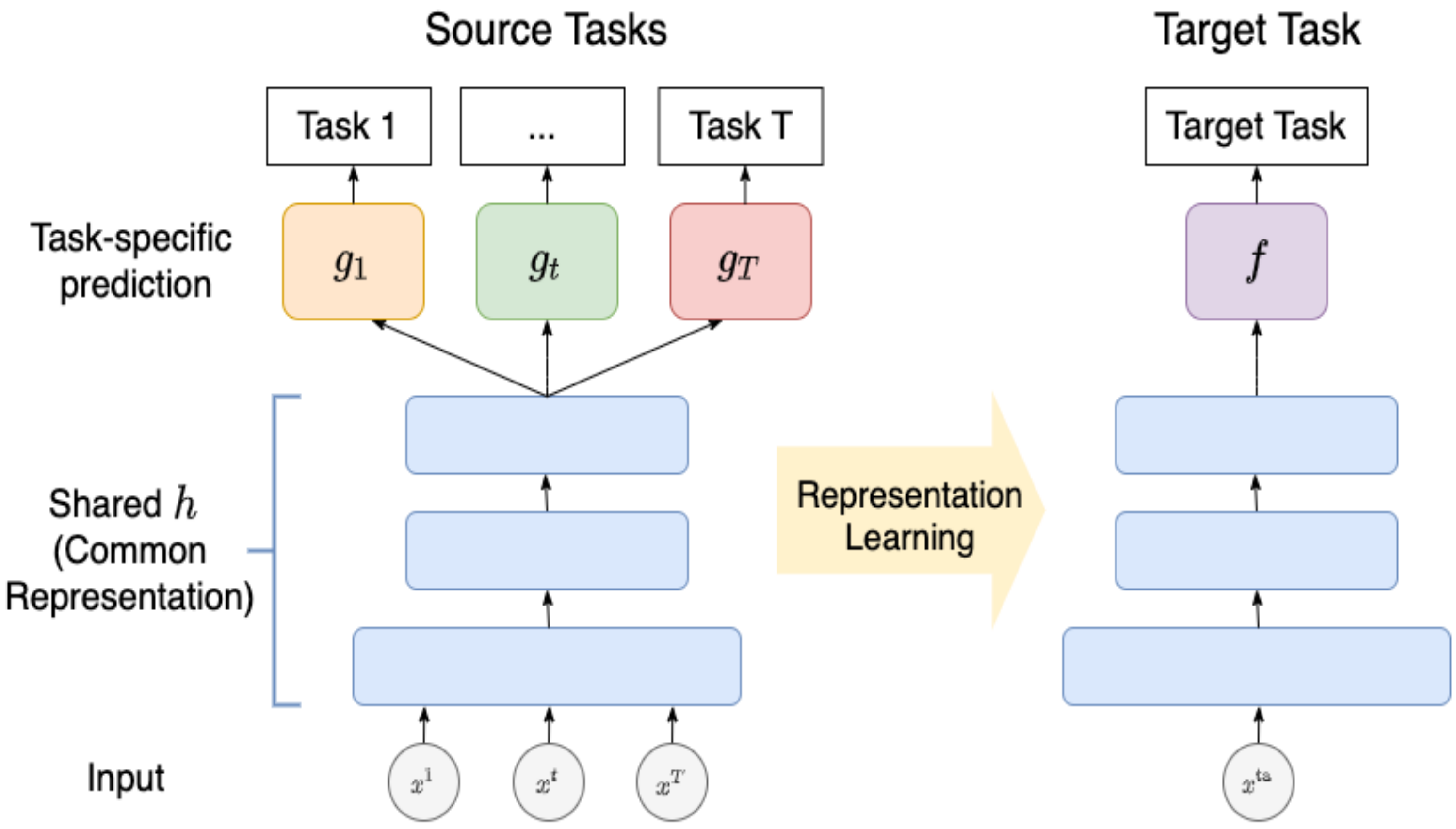
$$\min_{h\in\mathcal{H}} \sum_{t=1}^{T} \min_{g_t\in\mathcal{G}} \sum_{i=1}^{n_1} \ell\left(g_t\left(h(\boldsymbol{s}_i^t)\right), \boldsymbol{\pi}^*(\boldsymbol{s}_i^t)\right)$$

$\ell$: loss function

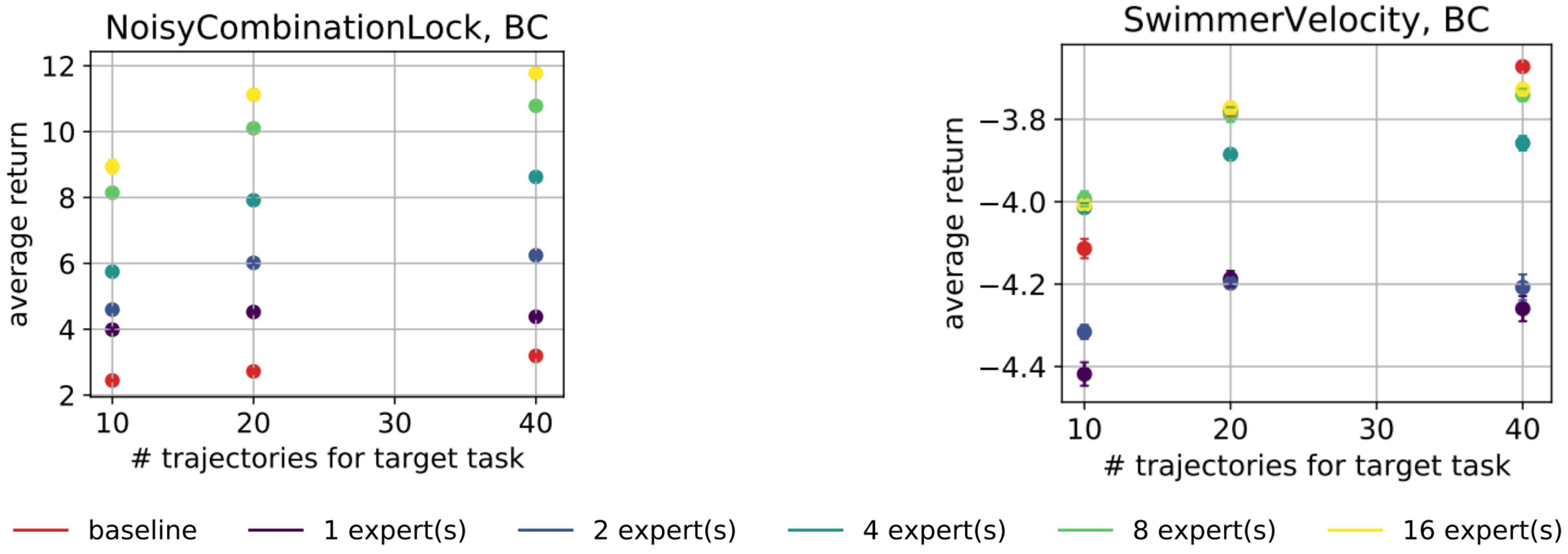
Training for target task:

$$\min_{f\in\mathcal{G}} \sum_{i=1}^{n_2} \ell\left(f\left(h(\boldsymbol{s}_i^{ta})\right), \boldsymbol{\pi}^*(\boldsymbol{s}_i^{ta})\right)$$

Representation $h(\cdot)$ is fixed



# Experiments




Control the agent towards a target location

Control the agent with a target velocity (MuJoCo)

**Representation learning helps:**

- Beats the baseline for small $n_2$ (# trajectories for target task).
- Increasing # of source tasks (experts) helps.

[Arora D. Kakade Luo Saunshi ICML 2020]

# Summary

**When and Why Does Representation Learning Help?**

- When: existence of a good representation & diversity of source tasks.
- Why: joint optimization forces to learn a good representation.
- **Open Problem:** optimization theory for representation learning.

**Representation Learning for Other Settings:**

- Imitation learning.
- **Future directions:** reinforcement learning? control?

# Thank You

# Two-layer Over-parameterized NN

$\mathcal{H}$: ReLU neural networks. $h(x) = \sigma(Bx)$.
$x \in \mathbb{R}^d, B \in \mathbb{R}^{k\times d}$ (k very large), $\sigma$: ReLU.

**Assumption 1: Existence of a Good Representation**

There exist a linear representation $B^* \in \mathbb{R}^{k\times d}$, and $w_1^*, w_2^*, \dots, w_T^*, w_{ta}^* \in \mathbb{R}^k$:

$$\mathbb{E}_{(x_t,y_t)\sim\mu_t}[\ell(\langle w_t^*, \sigma(B^* x_t)\rangle, y_t)] = 0 \;\forall t = 1, \dots, T$$

$$\mathbb{E}_{(x_{ta},y_{ta})\sim\mu}[\ell(\langle w_{ta}^*, \sigma(B^* x_{ta})\rangle, y_{ta})] = 0$$

**Assumption 2: Diversity of Source Tasks for Linear Predictor**

$w_{ta}^*$ is contained in the span of $W^* = [w_1^*, w_2^*, \dots, w_T^*] \in \mathbb{R}^{k\times T}$.

The optimal predictor of the target task is covered by the those of source tasks.

# Main Result for Two-layer Over-parameterized NN

**Theorem [Du Hu Kakade Lee Lei, 2020]**

Under Assumption 1 &2, we have

$$\mathbb{E}_{(x^{ta},y^{ta})\sim\mu}\left[\ell\left(f\left(h(x^{ta})\right),y^{ta}\right)\right] = O\left(\frac{tr(\Sigma)}{\sqrt{n_1 T}} + \frac{||\Sigma||_{op}}{\sqrt{n_2}}\right)$$

where $\Sigma$ is the covariance of input $x$

Without representation learning, directly learning with a two-layer over-parameterized neural network: $O(\frac{tr(\Sigma)}{\sqrt{n_2}})$.