Self-Supervised Learning

Joseph Bakarji

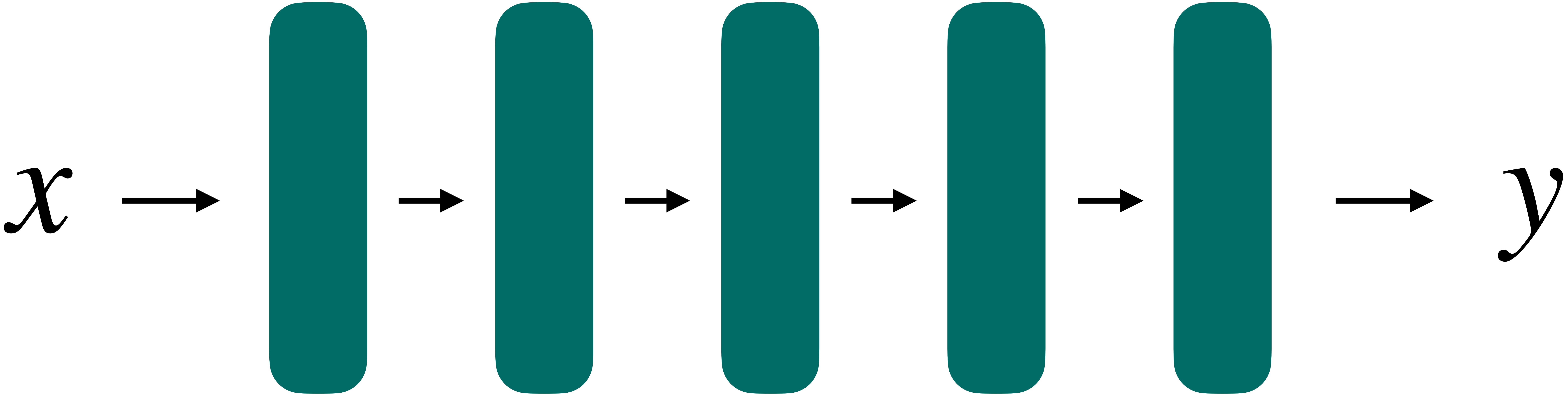
Supervised Learning

Supervised Learning (Training)

$$x \longrightarrow \boxed{f} \longrightarrow y$$

Transfer Learning

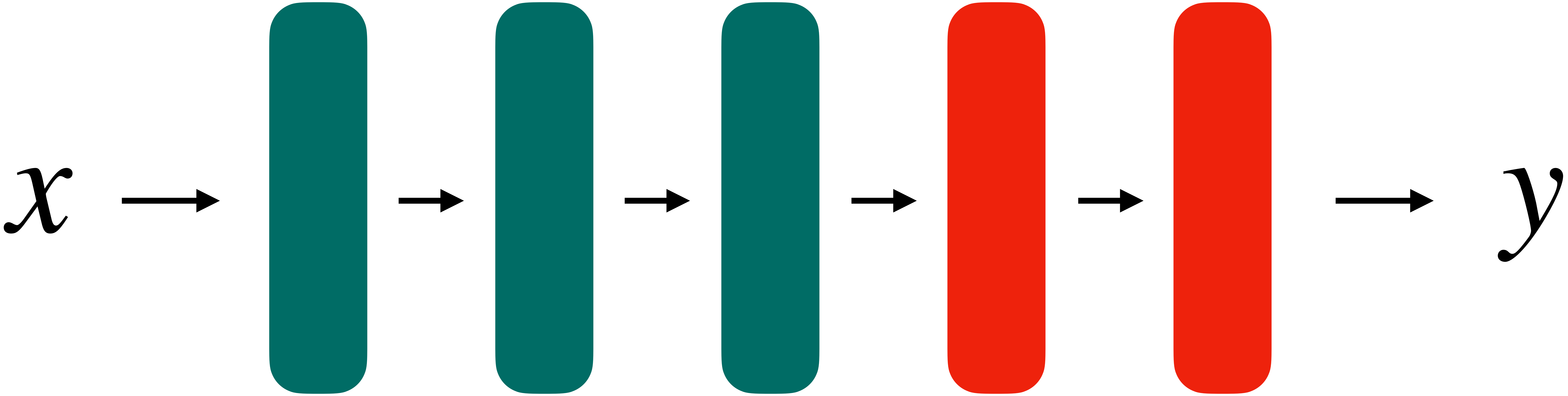
Train Model on Big Dataset of Images



$$f_1 \circ f_2 \circ f_3 \circ f_4 \circ f_5$$

Transfer Learning

Retrain model on your data and task
By changing the 'head' of the network



$$f_1 \circ f_2 \circ f_3 \circ g_4 \circ g_5$$

A Paradigm Shift

Self-Supervised Learning

Pre-Training

$$x \rightarrow \boxed{\phi_\theta}$$

Pre-train a large models on a large scale unlabeled dataset

Adaptation

$$x \rightarrow \boxed{f} \rightarrow y$$

Adapt the retrained model to a wide range of tasks

Self-Supervised Learning

Pre-Training

$$x \rightarrow \phi_\theta$$

Pre-train a large models on a large scale unlabeled dataset

- Data: $\{x^{(1)}, \ldots, x^{(n)}\}$
- Feature Map: $\phi_\theta(x) \in \mathbb{R}^m$ (learned in some cases)
- Pre-training Loss

$$L_{pre}(\theta) = \frac{1}{n}\sum_{i=1}^{n} l_{pre}(\theta, x^{(i)})$$

- Optimize $L_{pre}(\theta) \rightarrow \hat{\theta}$

Self-Supervised Learning

Adaptation

$$x_t \longrightarrow \boxed{f} \longrightarrow y_t$$

Adapt the retrained model to a wide range of tasks

- Data: $\{(x_t^{(1)}, y_t^{(1)}), \ldots, (x_t^{(n_t)}, y_t^{(n_t)})\}$
- $n_t = 0$: zero-shot learning
- n_t is small: few-shot learning

Adaptation: Linear Probe/Head

Adaptation

$$x_t \rightarrow f \rightarrow y_t$$

Adapt the retrained model
to a wide range of tasks

Prediction Model

$$f = w^\top \phi_{\hat{\theta}}(x)$$

Train w with loss function

$$\min_{w} \frac{1}{n_t} \sum_{i=1}^{n_t} l_{task}(y_t^{(i)}, w^\top \phi_{\hat{\theta}}(x_t^{(i)}))$$

Adaptation: Finetuning

Adaptation

$$x_t \longrightarrow \boxed{f} \longrightarrow y_t$$

Adapt the retrained model
to a wide range of tasks

Prediction Model

$$f(w, \theta) = w^\top \phi_{\hat{\theta}}(x)$$

Optimize both w, θ on downstream task

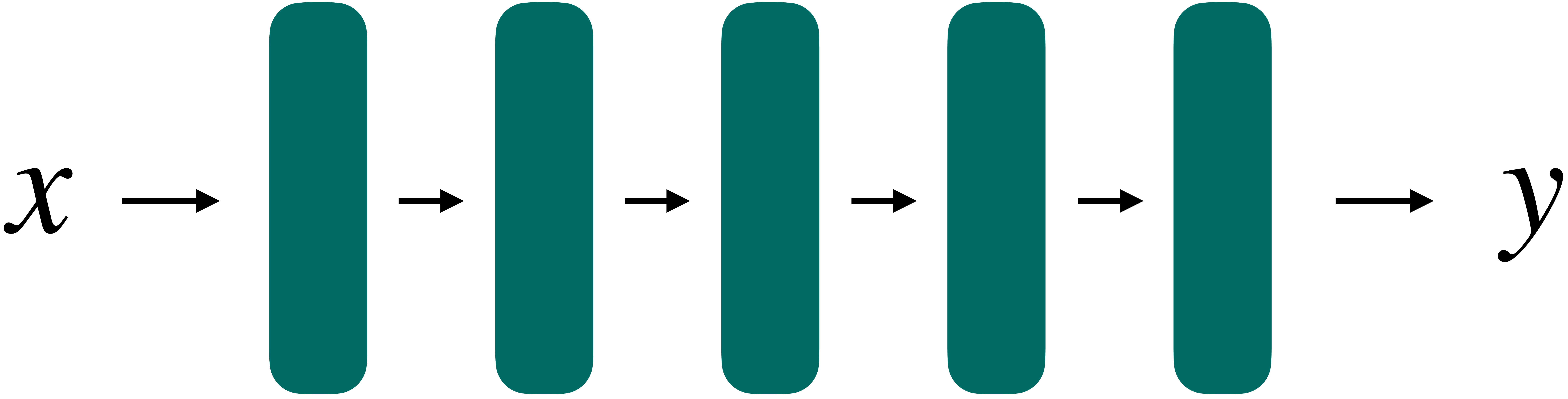
$$\min_{w,\theta} \frac{1}{n_t} \sum_{i=1}^{n_t} l_{task}(y_t^{(i)}, w^\top \phi_{\hat{\theta}}(x_t^{(i)}))$$

Initialize

$\theta \leftarrow \hat{\theta}$ and $w \leftarrow$ random

Computer Vision: supervised pretraining

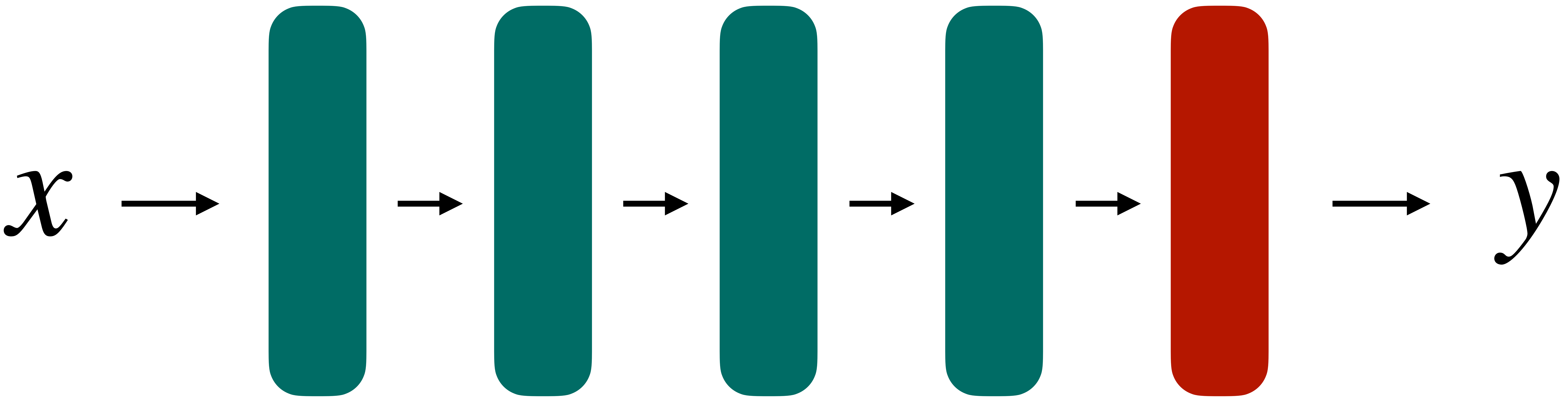
Train Model on Big Dataset of Images



$$f_1 \circ f_2 \circ f_3 \circ f_4 \circ f_5$$

Computer Vision: supervised pretraining

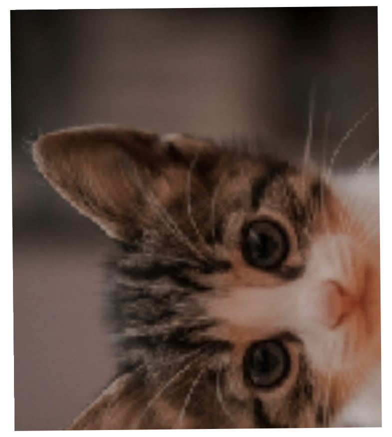
Train linear head on last layer



$$f_1 \circ f_2 \circ f_3 \circ f_4 \circ g_5$$

Constrastive Learning (no labels)

- **Data Augmentation**
  - Image random crop
  - Flip
  - Color Transformation

$\hat{x}$ $\tilde{x}$ $\hat{z}$

$$\begin{tikzcd} & \hat{x} \\ x \arrow[ur] \arrow[dr] & \\ & \tilde{x} \end{tikzcd}$$

Make $\phi_\theta(\hat{x})$ and $\phi_\theta(\tilde{x})$ have similar representations

$\hat{x}$
$\hat{z}$

Random/negative Pair

Make $\phi_\theta(\hat{x})$ and $\phi_\theta(\hat{z})$ far from each other

Constrastive Learning (no labels)

$$\begin{tikzcd} & \hat{x} \\ x \arrow[ur] \arrow[dr] & \\ & \tilde{x} \end{tikzcd}$$

Make $\phi_\theta(\hat{x})$ and $\phi_\theta(\tilde{x})$ have similar representations

Random/negative Pair

$$\hat{x} \quad \hat{z}$$

Make $\phi_\theta(\hat{x})$ and $\phi_\theta(\hat{z})$ far from each other

SIMCLR Loss Function

$$x^{(1)} \quad x^{(2)} \quad \dots \quad x^{(B)}$$

$$\hat{x}^{(1)} \quad \hat{x}^{(2)} \quad \dots \quad \hat{x}^{(B)}$$

$$\tilde{x}^{(1)} \quad \tilde{x}^{(2)} \quad \dots \quad \tilde{x}^{(B)}$$

$$-\sum_{i=1}^{B} \log \frac{\exp\left(\phi_\theta(\hat{x}^{(i)})^\top \phi_\theta(\tilde{x}^{(i)})\right)}{\exp\left(\phi_\theta(\hat{x}^{(i)})^\top \phi_\theta(\tilde{x}^{(i)})\right) + \sum_{j \neq i} \exp\left(\phi_\theta(\hat{x}^{(i)})^\top \phi_\theta(\tilde{x}^{(j)})\right)}$$

$$-\sum_{i=1}^{B} \log \frac{A}{A+B}$$

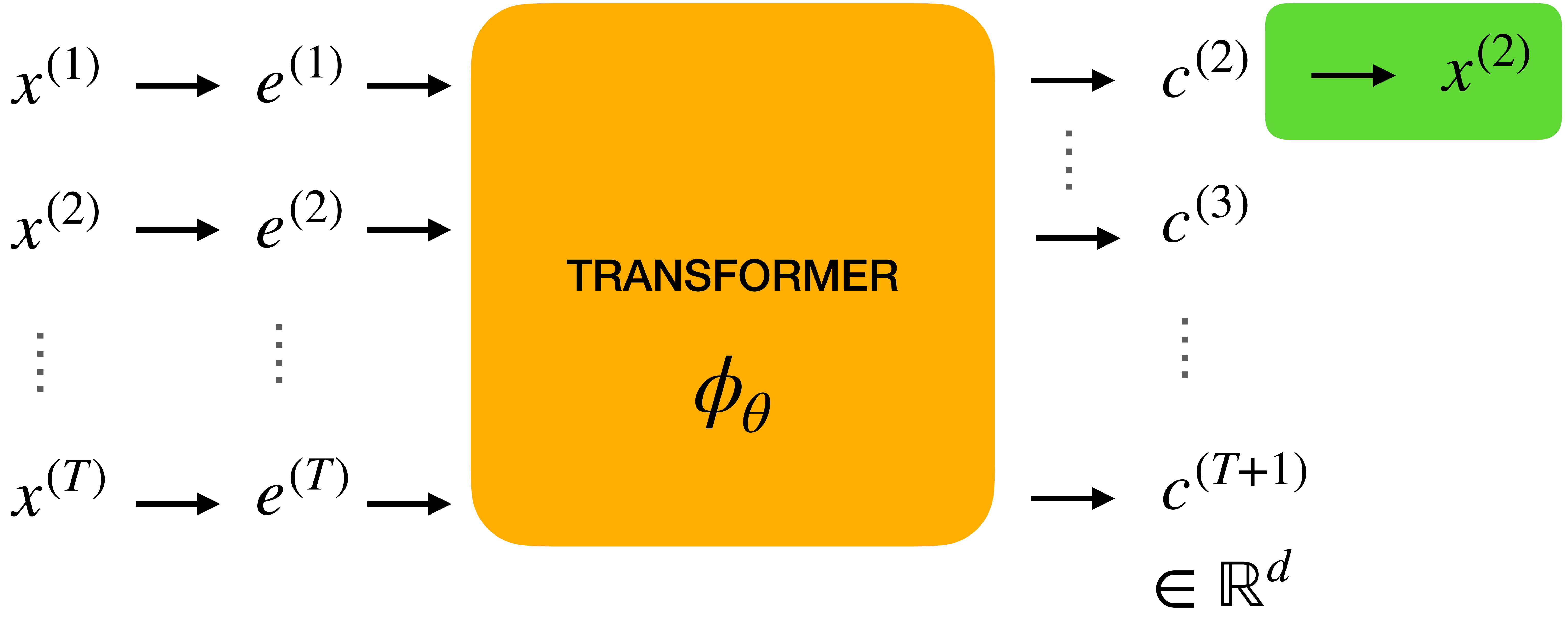
We want **B** to be small and **A** to be large

Large Language Models

- Given a series of words, $(x^{(1)}, \ldots, x^{(T)})$ in a vocabulary: $x^{(i)} \in \{1,\ldots,V\}$
- **A Language Model** is a probabilistic model for $p(x^{(1)}, x^{(2)}, \ldots, x^{(T)})$
- Use **Chain Rule:**
    - $p(x^{(1)}, x^{(2)}, \ldots, x^{(T)}) = p(x^{(1)})\ p(x^{(2)} | x^{(1)})\ p(x^{(3)} | x^{(1)}, x^{(2)}) \ldots$
- **Model:**
    - $p(x^{(t)} | x^{(1)}, \ldots, x^{(t-1)})$ which is V dimensional instead of V^T dimensional

Large Language Models

- **Embedding** (learned) for each word $x^{(i)}$ into a vector $e_i \in \mathbb{R}^d$



Use it to predict

$$p(x^{(t)} \,|\, x^{(1)}, \ldots, x^{(t-1)})$$

$$= \begin{bmatrix} p(x^{(t)} = 1 \mid x^{(1)}, \ldots, x^{(t-1)}) \\ \vdots \\ p(x^{(t)} = V \mid x^{(1)}, \ldots, x^{(t-1)}) \end{bmatrix}$$

$$= \text{Softmax}(w^{(t)} \cdot c^{(t)}) \quad \in \mathbb{R}^V$$

$$c^{(t)} \in \mathbb{R}^d \qquad w^{(t)} \in \mathbb{R}^{V \times d}$$