Provable Representation Learning

Jason D. Lee

Princeton University

Joint work with

Qi Lei, Simon Du, Wei Hu, Sham Kakade, Nikunj Saunshi, and Jiacheng Zhuo.

https://arxiv.org/abs/2002.09434

https://arxiv.org/abs/2008.01064

Outline

Representation Learning

Representation Learning

Deep Learning's success is due to learning useful representations.

- Deep networks **learns the feature representation**.
- Feature representations transfer to other tasks.
- Representations can be learned on unlabeled data.
- Competing methods lack transferrability.

Competing methods are unable to do this (random forests, kernel machines, gradient boosting)

How is deep representation learning done?

Simplest algorithm:

1. Train a deep network on some task (can be on Imagenet or self-supervised task).
2. Keep only the body, and discard the head.
3. Retrain (or finetune) the head using labeled data from target domain.

Imagenet (Supervised) Pretraining

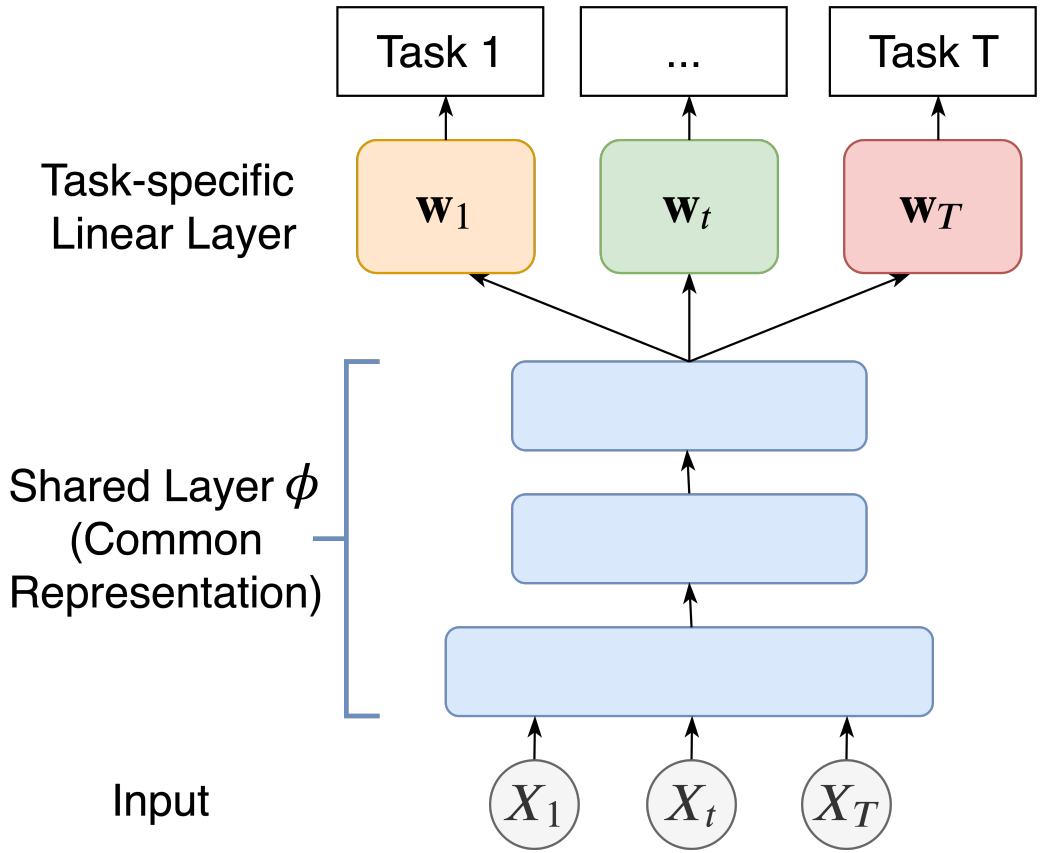
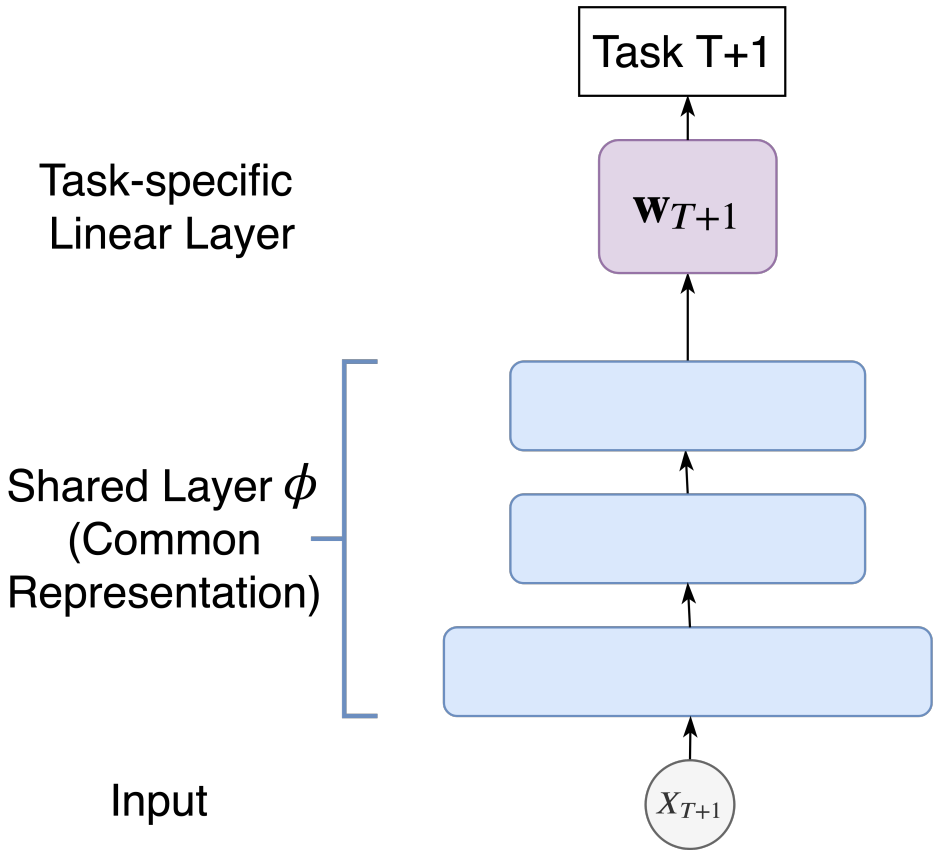


Figure: Dates back to at least Caruana 1997.

Representation Learning: Target Task



Train new head using labeled data from target task.

Applications of Representation Learning

Many of the DL success stories:

- Any domain without enough data.
- Even applications with a lot of labeled data (Imagenet) can benefit from self-supervised pretraining.
- Language models.
- Transfer learning.
- Meta-learning
- Reinforcement Learning
- $\vdots$

In Theory?

Framework.

$f(x) = g \circ \phi(x)$ with $g \in \mathcal{G}, \phi \in \Phi$.

- T tasks, n_S samples per source task, and n_T samples on the target task. $\mathcal{C}(\mathcal{F})$ is complexity.
- Maurer and Baxter proves results of the type:

$$\text{Risk} \lesssim \frac{\mathcal{C}(\Phi)}{\sqrt{T}} + \frac{\mathcal{C}(\mathcal{G})}{\sqrt{n_T}}.$$

- Cannot use the n_S samples across tasks to learn the representation.

Ideal Representation Learning

Goal

Under natural assumptions,

$$\text{Risk} \lesssim \frac{\mathcal{C}(\Phi)}{n_S T} + \frac{\mathcal{C}(\mathcal{G})}{n_T}.$$

Without these assumptions, such a rate is not attainable.

Ideal Representation Learning

Comparison

$$\frac{\mathcal{C}(\Phi)}{\sqrt{T}} + \frac{\mathcal{C}(\mathcal{G})}{\sqrt{n_T}} \text{ vs } \frac{\mathcal{C}(\Phi)}{n_S T} + \frac{\mathcal{C}(\mathcal{G})}{n_T}$$

- Biggest gain: pool all $n_S \times T$ samples to learn the complex representation $\phi \in \Phi$.
- Minor gain: slow rate improved to fast rate.
- Old bound does not depend on n_S. The same rate for $n_S = 1$ and $n_S = 10^8$.

How do attain ideal rate

Assumptions:

1. Shared good representation across tasks: $y_t = w_t^\top \phi^*(x)$.
2. Diversity of the $\{w_t\}$.

Why?

1. Shared representation encodes what transfers across the tasks.
2. Diversity of the $\{w_t\}$ (at least needs to "cover" w_{T+1}.)

Necessity of the Assumptions

Failure Cases

1. If ϕ^* is not shared across task, cannot obtain $\frac{\mathcal{C}(\Phi)}{n_S T}$.
2. If w_{T+1} is in a direction not spanned by w_t, then you have not learned ϕ^* in that direction.

Algorithm

For Source Tasks:

$$\hat{\phi} \leftarrow \underset{\phi \in \Phi, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_T \in \mathbb{R}^k}{\text{minimize}} \frac{1}{2n_1 T} \sum_{t=1}^{T} \left\| \boldsymbol{y}_t - \phi(X_t) \boldsymbol{w}_t \right\|^2 .$$

For Target Task: train a linear predictor on top of $\hat{\phi}$:

$$\hat{\boldsymbol{w}}_{T+1} \leftarrow \underset{\boldsymbol{w}_{T+1} \in \mathbb{R}^k}{\text{minimize}} \frac{1}{2n_2} \left\| \boldsymbol{y}_{T+1} - \hat{\phi}(X_{T+1}) \boldsymbol{w}_{T+1} \right\|^2 .$$

Warmup: Linear Representation Learning

Notation:

- $\phi(x) = Bx$ for $B = k \times d$ encodes k-dimensional subspace.
- $y_t = \theta_t^\top x = w_t^\top Bx$, so all θ_t live in the **same** k-dimensional subspace.

Assumptions:

- Shared representation: Same B for every task.
- Diversity: Stack $\{w_t\}$ into a matrix W and the matrix has condition number $O(1)$.

Linear Representation Learning

Theorem

Under these assumptions,

$$\text{Risk} \lesssim \frac{kd}{n_S T} + \frac{k}{n_T}.$$

- *No dependence on condition numbers: operating in a regime where span(B) is not estimable.*
- *Extends to the case of covariate shift. Does not need $p_t(x)$ shared.*
- *Concurrent work by Tripuraneni et al. for isotropic Gaussian x (no covariate shift and span(B) is estimable).*
- *Algorithmic implementable via Nuclear Norm relaxation, but costs condition numbers in the risk.*

General Representation Learning

Theorem

Under the same assumptions,

$$\textit{Risk} \lesssim \frac{\mathcal{C}(\Phi)}{n_S T} + \frac{k}{n_T}.$$

- We do not know how to implement in polynomial time.
- Covariate shift is allowed.

Two-layer Neural Network Representations

- Source Tasks with Weight decay:

$$\hat{B}, \hat{W} = \underset{\substack{B \in \mathbb{R}^{d_0 \times d}, \\ W=[\boldsymbol{w}_1, \cdots, \boldsymbol{w}_T] \in \mathbb{R}^{d \times T}}}{\arg\min} \frac{1}{2n_1 T} \sum_{t=1}^{T} \|\boldsymbol{y}_t - (X_t B)_+ \boldsymbol{w}_t\|^2 + \frac{\lambda}{2}\|B\|_F^2 + \frac{\lambda}{2}\|W\|_F^2.$$

- Training on target task:

$$\hat{\boldsymbol{w}}_{T+1} \leftarrow \underset{\|\boldsymbol{w}\| \leq r}{\arg\min} \frac{1}{2n_2} \|\boldsymbol{y}_{T+1} - (X_{T+1}\hat{B})_+ \boldsymbol{w}\|^2.$$

Neural Network Excess Risk

Theorem

$$\textit{Risk} \leq \frac{\textit{Rademacher}}{\sqrt{n_1 T}} + \frac{\|w^*\|}{\sqrt{n_T}}$$

No labeled data?

Create your own labels

Supervised pretraining needs labels from related tasks. What if this isn't available?

Create labels from the input data.

1. Supervised Representation Learning

2. Self-Supervised Learning

Self-Supervised Representation Learning

Self-Supervised Learning.

Predict functions of the input from other parts.

- Denoising autoencoder.
- Context Encoder (Image inpainting)
- Next word prediction and missing word prediction.
- Image colorization.

⋮

Learning Pikachu

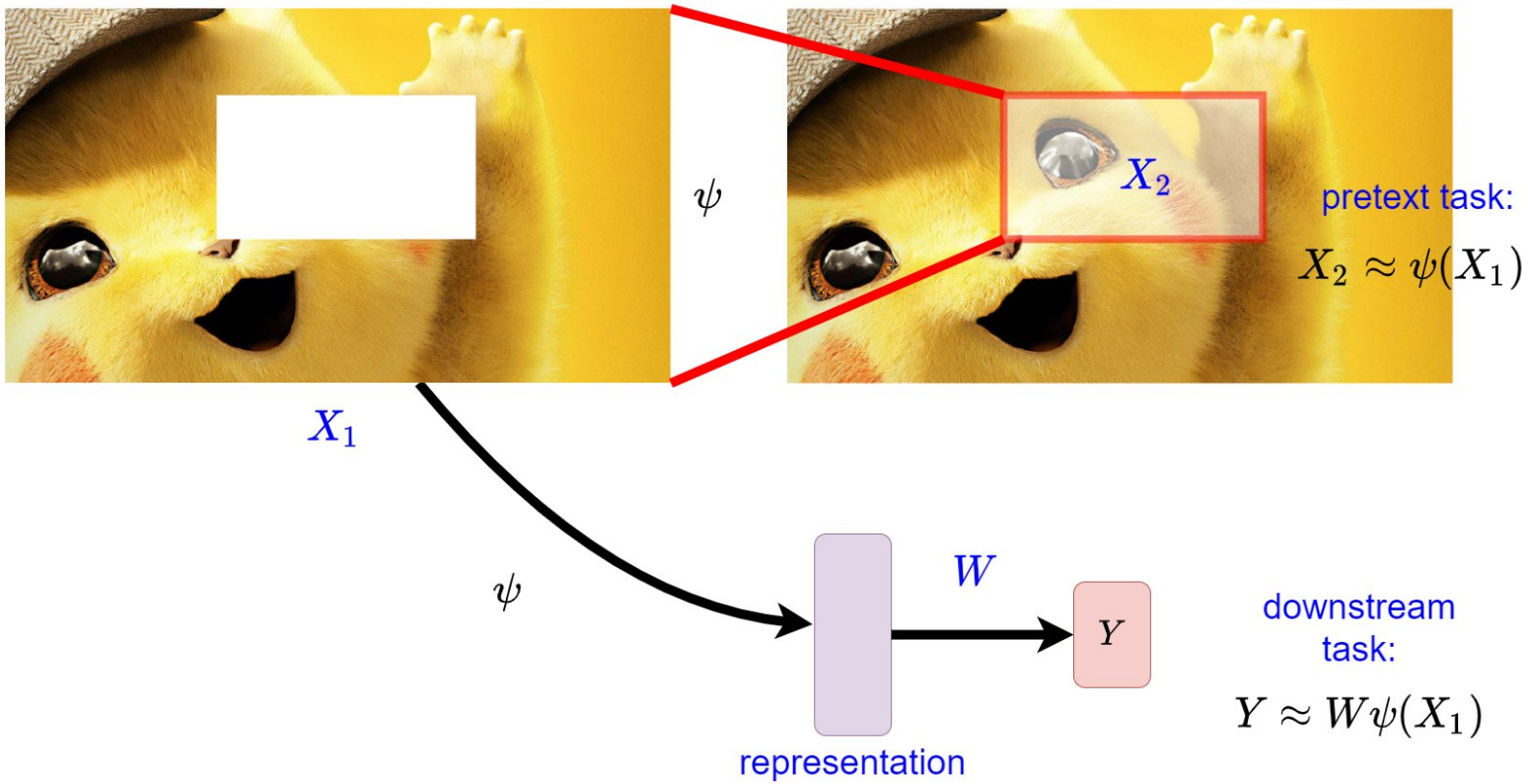


Figure: Self-supervised Learning: Predicting what you already know.

Self-supervised Learning: Does it work?

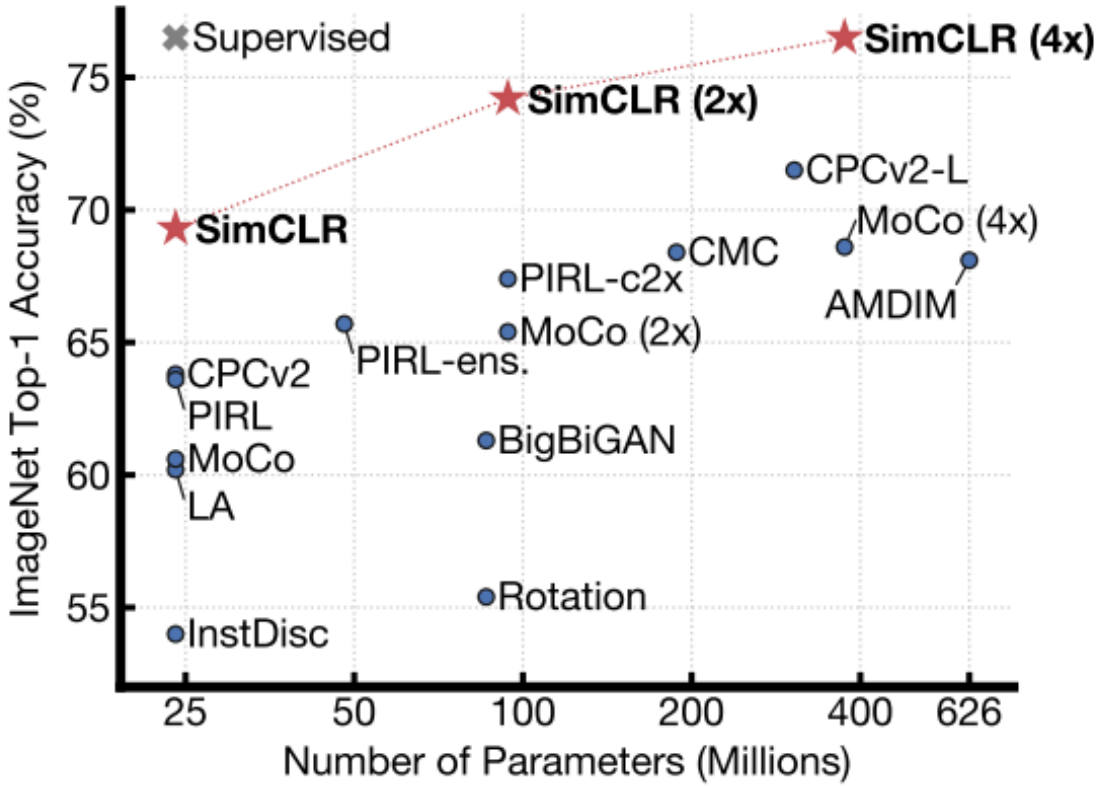


Figure: Linear classifier + pretraining on pretext task outperforms supervised learning. Credit: Google AI blog.

Self-Supervised for Transfer

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
SimCLR (ours)	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned:												
SimCLR (ours)	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5

Figure: Pretrained on ImageNet via self-supervised learning. (Credit: SimCLR)

Why?

Why?

Why does predicting parts of the input help?

- No extra information, we already observed the entire input.
- Hope: ψ keeps the relevant parts of $y|x$ such that a simple classifier (e.g. linear or fine-tuned)

Prior Related Work.

1. Contrastive Learning when have access to $x^+, x \sim C^+$ and $x^- \sim C^-$ (Arora et al.) learns a mean classifier.
2. Contrastive Learning in topics models (Tosh et al., Daniel's talk) learns posterior distribution and multview model.
3. Multiview learning/Redundancy (Foster and Kakade) when both x_1, x_2 get low error. Algorithm uses CCA to reduce dimension.
4. Information bottleneck "explanations".

This talk.

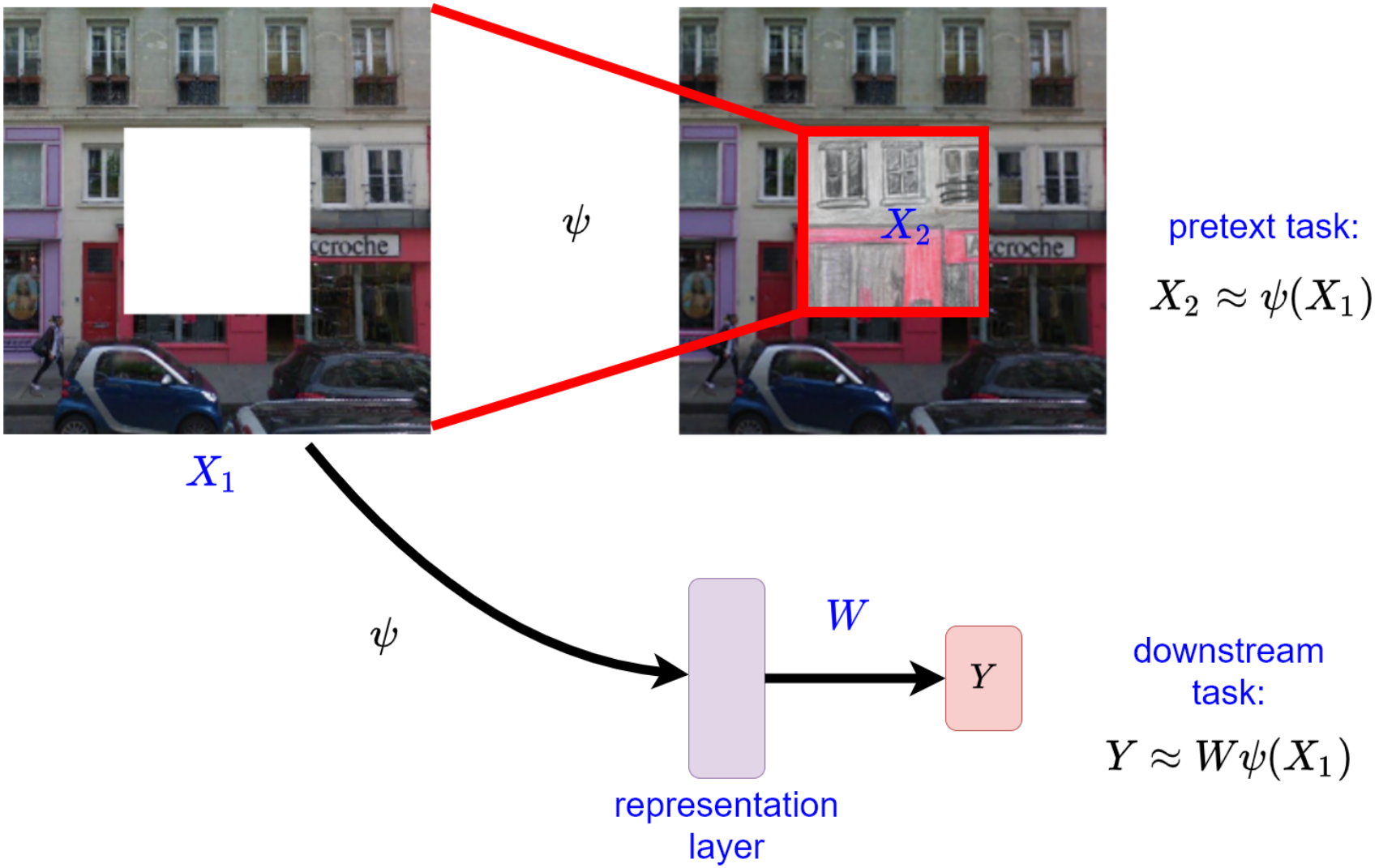


Figure: Context Encoder (Pathak et al.)

Algorithm

1. Take input and partition into (x_1, x_2) (e.g. x_2 is small patch and x_1 is remainder of the image).
2. Setup pretext task as $\min_\psi \mathbb{E}\|X_2 - \Psi(X_1)\|^2$. Typically $\psi(x_1) = D_\phi \circ E_\theta(x_1)$ for a deep network D_ϕ, E_θ.
3. Use n_L labeled samples to learn

$$\min_W \|Y - \psi(X_2)W\|^2.$$

Variants:

- Choose ψ as only encoder E_θ.
- Solve many different pretext tasks.

Notation

1. Label $Y \in \mathbb{R}^k$.
2. $X_1 \in \mathbb{R}^{d_1}$ and $X_2 \in \mathbb{R}^{d_2}$.
3. Latent variables $Z \in \mathbb{R}^m$.

Warm-up: Linear case.

1. $\psi(x_1) := Bx_1$, where $B = \arg\min \mathbb{E}\|X_2 - BX_1\|^2 \in \mathbb{R}^{d_1 \times d_2}$.
2. Learn $w = \arg\min \mathbb{E}\|Y - w^\top BX_1\|^2$.

Compare this to $\theta = \arg\min \mathbb{E}\|Y - \theta^\top X_1\|^2$. In the linear case, there is closed-form:

$$\theta = Bw^* + \Sigma_{X_1, X_2|Y}\delta.$$

1. If $X_1 \perp X_2|Y$ (partial correlation is 0), then $\theta = Bw^*$.
2. Only need k (dimension of Y) samples instead of d_1.

Warm-up: Linear Case

What if $\Sigma_{X_1,X_2|Y}$ is big?

- Consider some latent variables Z such that $X_1, X_2 \mid Y, Z$ (or almost partially uncorrelated)

$$\text{Risk} \leq \mathcal{O}\left(\frac{k+m}{n_L} + \frac{1}{\beta}\|\Sigma_{X_1,X_2|Y,Z}\| + \epsilon_{\text{pre}}\right),$$

- ϵ_{pre} is accuracy of learning pretext task.
- Sample complexity reduced from $n \asymp d$ to $n \asymp k+m$.
- $\beta = \sigma_{k+m}(\Sigma_{\bar{Y}\bar{Y}}^{-1}\Sigma_{\bar{Y}X_2})$, X_2 and $\bar{Y}$ cannot be uncorrelated.

Characterizing Conditional Independence

Define $\epsilon_{\mathsf{CI}} = \mathbb{E}_{X_1} \left\| \mathbb{E}[X_2|X_1] - \mathbb{E}_{Y,Z}[\mathbb{E}[X_2|Y,Z]|X_1] \right\|^2$. This is 0 if $X_1, X_2 \mid Y, Z$.

- Hilbert-Schmidt norm of a certain conditional cross-covariance operator.

$$\text{Excess Risk} \leq \mathcal{O}\left(\frac{k+m}{n_L} + \epsilon_{\mathsf{CI}} + \epsilon_{\mathsf{pre}} \right).$$

- ERM would need $n \asymp$ Complexity of Function Class.
- X_2 does not linearly predict Y, but $\psi(X_2)$ does.
- Excess risk is relative to the best predictor $f^*(X_1)$ over all functions f.

Future Directions.

- Provable **algorithms** for representation learning (preferably SGD).
- Approximate conditional independence is not generally applicable. What other conditions allows self-supervised learning to work? Design pretext tasks motivated by theory.
- Contrastive loss vs reconstruction loss?
- Finetuning, refine the representation when n_L is larger.

Thank you!