From amoebas to infants
Toward the evolution and development of prior knowledge

Phillip Isola, MIT
May 2019
The big goal: knowledge that generalizes

We want agents with **prior knowledge** that allows them to learn quickly on new problems.
What kind of knowledge generalizes?

(Other things that may not generalize as well: direct perception, expert knowledge, specific policies, extrinsic goals, …)
Intelligent agents with knowledge that generalizes
Intelligent agents with knowledge that generalizes
Intelligent agents with knowledge that generalizes
Sensory representation learning
Representation Learning

\[ X \rightarrow \text{compressed image code (vector } z) \]
Representation Learning

Image $X$ compressed image code (vector $z$) Reconstructed image $\hat{X}$

[e.g., Hinton & Salakhutdinov, Science 2006]
Reconstructed image

Image

\[ \hat{X} = \mathcal{F}(X) \]

[e.g., Hinton & Salakhutdinov, Science 2006]
Data compression

$X \xrightarrow{} \hat{X}$

[Hinton & Salakhutdinov, Science 2009]
Data prediction

Some data \( \mathbf{X}_1 \) \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \rightarrow \right
Grayscale image: \( L \) channel

\[
X \in \mathbb{R}^{H \times W \times 1}
\]

Color information: \( ab \) channels

\[
\hat{Y} \in \mathbb{R}^{H \times W \times 2}
\]

[Zhang, Isola, Efros, ECCV 2016]
Grayscale image: L channel

\[ X \in \mathbb{R}^{H \times W \times 1} \]

Semantics? Higher-level abstraction?

\[ \mathcal{F} \]

information: ab channels

\[ \hat{Y} \in \mathbb{R}^{H \times W \times 2} \]

[Zhang, Isola, Efros, ECCV 2016]
Instructive failure
Instructive failure
Stimuli that drive selected neurons (conv5 layer)
“Autoencoder”

Input Image → $\mathcal{L}$ Grayscale $\xrightarrow{\mathcal{F}_1}$ Predicted Color $\xrightarrow{\mathcal{F}_2}$ Predicted Grayscale → Predicted Image
"Split-Brain Autoencoder"

Raw Data Channels

\[ X \]

\[ X_1 \]

\[ \hat{X}_2 \]

\[ F_1 \]

\[ F_2 \]

\[ \hat{X}_2 \]

\[ X_1 \]

\[ X \]

Predicted Data Channels

[Zhang, Isola, Efros, CVPR 2017]
Is the code informative about object class $y$?

Logistic regression:

$$y = \sigma(Wz + b)$$
Rather than compress, predict

Classification performance

Accuracy

Raw Data

Predicted Data

Reconstructed Data

Task from [Russakovsky et al. 2015]
Hypothesis:
Some bits are better than others
Useful bits are those shared between multiple sensory views

Max likelihood models:
A bit is a bit is a bit
**Contrastive Multiview Coding**

[Tian, Krishnan, Isola, under review]

**Hypothesis:**

Some bits are better than others

Useful bits are those shared between multiple sensory views

**Max likelihood models:**

A bit is a bit is a bit
Contrastive Predictive Coding
[van den Oord, Li, Vinyals, 2018]
\[ x = \{v^i_1, v^i_2\} \quad y = \{v^i_1, v^j_2\} \]
\[ x \sim p(v_1, v_2) \quad y \sim p(v_1)p(v_2) \]
\[ h_\theta(\{v_1, v_2\}) = e^{f_\theta_1(v_1)^TW_{12}f_\theta_2(v_2)} \]
\[ \mathcal{L}_{\text{contrast}} = -\mathbb{E}_S \left[ \log \frac{h_\theta(x)}{h_\theta(x) + \sum_{i=1}^k h_\theta(y_i)} \right] \]
\[ z_1 = f_\theta_1(v_1) \quad z_2 = f_\theta_2(v_2) \]
\[ I(z_1; z_2) \geq \log(k) - \mathcal{L}_{\text{contrast}} \]

[Tian, Krishnan, Isola, under review]
The parameterized statistical model is specified as a deep nonlinear transformation from views in latent space. The predictive approach has been extended to scenarios that include more than two views. We show its connection to mutual information between views and further extend it to scenarios that allow more efficient learning than the binary one, especially when the number of negatives, which is the case in our framework.

**3.1. Predictive Learning**

We generalize our CMC to different formulations: (1) "core view" specifies one view, and all other views are contrasted against that view; (2) "full graph" specifies all views, and mutual information between two views. We show its connection to mutual information between views and further extend it to scenarios that allow more efficient learning than the binary one, especially when the number of negatives, which is the case in our framework.

**3.2. Contrastive Learning with Two Views**

Contrastive learning uses pairwise Euclidean loss functions. The predictive approach has been extended to scenarios that include more than two views.

Formally, let $V_1, V_2, ..., V_m$ be samples from the data, denoted as $v$. Formally, let $y$ be samples from the noise distribution $z$. For $v_i$, we define the negative set to avoid collision.

We maximize the negative contrastive loss among samples following the negative distribution and predicting sound from vision with $\text{NCE}$.

Similarly to CPC, $\text{NCE}$ is a powerful contrastive approach that allows more efficient learning than the binary one, especially when the number of negatives, which is the case in our framework.

We maximize the negative contrastive loss among samples following the negative distribution and predicting sound from vision with $\text{NCE}$.

Transfer performance on STL-10 (linear):

- 72%
- 83%

[Tian, Krishnan, Isola, under review]
We evaluate the representation of each view by contrasting its L channel representation, which is learnt using the patch-based contrastive loss and all baselines. As shown in Table 2, our CMC framework performs well compared to predictive learning. Though in the unsupervised stage we only use 1.3K images, contrastive learning framework consistently outperforms predictive learning [4, 5].

We then evaluate the representation quality by training a linear decoder on the learned representations. A number of empirical results show that our contrastive loss and all baselines, which serve as lower and upper bounds, as shown in Table 3, which shows our CMC produces high quality representations towards that of supervised ones, for all of views.

We compare CMC with the semantic labels from only the representation of L channel. We therefore investigate our CMC framework beyond L channel. To treat the quality of representations for all views, we therefore train it jointly with the decoder. This end-to-end optimization on multiple views or modalities should improve the performance of the fully supervised representations. A number of empirical results show that our contrastive learning framework performs well compared to predictive learning in this scenario where both the task and the dataset are unknown. We also include “random” and “supervised” baselines similar to that in previous sections.

While experiments in section 4.1 show that contrastive learning improves the accuracy, whenever geometry or semantic labels are considered, Table 6 shows that other modalities also benefit from contrastive learning.

**Table 6:** Performance on the task of using single view to predict semantic labels from other views. We compare CMC with baselines. As shown in Table 4, which shows our CMC produces high quality representations towards that of supervised ones, for all of views.

<table>
<thead>
<tr>
<th>Views</th>
<th>Predictive</th>
<th>Contrastive</th>
</tr>
</thead>
<tbody>
<tr>
<td>L, Depth</td>
<td>55.5</td>
<td><strong>58.3</strong></td>
</tr>
<tr>
<td>L, Normal</td>
<td>58.4</td>
<td><strong>60.1</strong></td>
</tr>
<tr>
<td>L, Seg. Map</td>
<td>57.7</td>
<td><strong>59.2</strong></td>
</tr>
<tr>
<td>Random</td>
<td>25.2</td>
<td></td>
</tr>
<tr>
<td>Supervised</td>
<td>65.1</td>
<td></td>
</tr>
</tbody>
</table>

[Tian, Krishnan, Isola, under review]
The idea behind contrastive learning is learning by discriminating, or comparing between samples from different distributions. We learn a score function \( \mathcal{L} \) that assigns a positive score to pairs of positive samples and a negative score to pairs of negative samples. Formally, \( \mathcal{L} = \sigma(x - y) \), where \( \sigma \) is an exponential function. At test time, \( \mathcal{L} \) is used to compute the distance between instances in an encoding space, and to correctly select a single positive sample.

\[
\mathcal{L}_C = \sum_{j=2}^{M} \mathcal{L}_{\text{contrast}}(V_1, V_j)
\]

\[
\mathcal{L}_F = \sum_{1 \leq i < j \leq M} \mathcal{L}_{\text{contrast}}(V_i, V_j)
\]

Contrastive Learning vs. Predictive Learning. The predictive learning framework is designed to be used with any number of views and sequential data. Deep Infomax (DIM) is an example of a contrastive learning method for multi-view data that uses a deep nonlinear transformation from chrominance. We define the predictive learning vs. contrastive learning framework and apply CMC to the case where each color view is predicted through linear projection of luminance and the product of marginals. We also apply CMC to the case where each view is predicted through linear projection of other views. We consider two practical implementations of the CMC framework: (1) "core view" specifies one view, and all other views are disfavored as negatives, i.e., \( \theta_i = \theta_1 \), and samples from the product of marginals, \( \mathcal{L}_F \), are used to store latent features for each data sample. Therefore, this method might be the luminance of a particular image and builds a representation for the color space.

We now put recent related work in the above framework. Such formulations offer different tradeoffs between efficiency and effectiveness. CMC signifi- cantly outperforms methods such as SplitBrain that use predictive learning; or BiGAN that use adversarial learning. CMC significantly outperforms SieGan and prior methods such as SplitBrain that use predictive learning. In particular, we find that CMC achieves better results with larger patch sizes and fewer negative samples. To avoid very large batch sizes, we consider two practical implementations introduced by Deep Infomax (DIM) (Oord et al., 2017; Hjelm et al., 2018) and Gutmann & Hyvärinen (2010).
To measure the quality of the learned representation, we consider the task of predicting semantic labels from the representation of \( \Phi \). Such as semantic segmentation where input size changes. For these tasks, we discard the fully connected layers and evaluate the spatial network from \( \Phi \) as a backbone network.

Setup. To extract features from each view, we use a neural network with \( 5 \) convolutional layers, and \( 1 \) fully connected layer. As the size of the dataset is relatively small, we adopt the patch-based contrastive objective to increase the number of negative pairs. Patches with a size of \( 128 \times 128 \) are added, both these metrics steadily increase. The views are (in order from the core view to the peripheral views): front, top, left, right, back. We use batch size of \( 32 \) and learning rate of \( 0.001 \) which is decayed by a factor of \( 5 \) after \( 200 \) and \( 250 \) epochs. CMC is trained with Adam for \( 300 \) epochs, with an initial learning rate of \( 0.001 \) which is decayed by a factor of \( 5 \) after \( 200 \) and \( 250 \) epochs.

We apply the learned representation to the task of action recognition. The spatial network from \( \Phi \) is chosen as a random frame from a well-established paradigm for evaluating pre-trained RGB CaffeNets \([\text{Pre-training.}]\) and on HMDB-51, CMC is second-best in performance. Increasing the number of views of the data from dorsal streams during pre-training produces higher accuracy compared to a single view. Unifying both ventral and dorsal streams instead of one) provides a boost for UCF-101.

We use batch size of \( 32 \) and learning rate of \( 0.001 \) which is decayed by a factor of \( 5 \) after \( 200 \) and \( 250 \) epochs. CMC is trained with Adam for \( 300 \) epochs, with an initial learning rate of \( 0.001 \) which is decayed by a factor of \( 5 \) after \( 200 \) and \( 250 \) epochs. To evaluate the transferability of our RGB CaffeNet, we consider the task of predicting semantics (ab channel), depth, surface normal \([\text{Geometry \# of Views UCF-101 HMDB-51}]\) for extracting features from images and optimizing (ab channel), depth, surface normal \([\text{Geometry \# of Views UCF-101 HMDB-51}]\) is utilized to perform the segmentation task.
Development of intrinsic motivation
What we want:

- Differentiable, dense guidance signal
What we have:

Sparse, distant, non-differentiable returns
How to make training signal nicer?

• Value functions and TD methods — make signal more proximal

• Reward shaping — make signal dense and instructive

• Use a model — objective can be made differentiable

• Meta-optimize the reward function

[Ackley & Littman, Artificial Life 1991]
[Sorg, Singh, Lewis, NeurIPS 2010]
[Niekem, Barto, Spector 2010]
[Guo, Singh, Lewis, Lee, IJCAI 2016]
Evolved policy gradients
Meta-learning a surrogate loss to train an RL agent

[Houthooft, Chen, Isola, Stadie, Wolski, Ho, Abbeel, NeurIPS 2018]
Evolved policy gradients
Meta-learning a surrogate loss to train an RL agent

[Houthooft, Chen, Isola, Stadie, Wolski, Ho, Abbeel, NeurIPS 2018]
The loss looks at recent behavior and scores it (this is more general than a reward function, which only looks at current state)

\[ \pi^* = \arg \max_{\pi} \mathbb{E}_{\tau \sim \mathcal{M}, \pi} [R_{\tau}] \]

\[ L_{pg} = \mathbb{E} \left[ R_{\tau} \sum_{t=0}^{H} \log \pi(a_t | s_t) \right] \]

\[ L_{ac} = \mathbb{E} \left[ \sum_{t=0}^{H} A(s_t, a_t) \log \pi(a_t | s_t) \right] \]

\[ L = \mathbb{E} \left[ f_\phi(\{s_t, a_t, \pi(a_t | s_t)\}_{t=0}^{H}) \right] \]
Meta-learning: train a learning algorithm to do well on a distribution of tasks

Random physics, limb size, etc

Task 1:

Task 2:

Task 3:

\[ \phi^* = \arg \max_{\phi} \mathbb{E}_{M \sim p(M)} \mathbb{E}_{\tau \sim M, \pi_{\theta^*}} [R_{\tau}] \]

\[ \theta^* = \arg \min_{\theta} \mathbb{E}_{\tau \sim M, \pi_{\theta}} [L_{\phi}(\pi_{\theta}, \tau)] \]
Learning to hop backwards

Off-the-shelf RL algorithm (PPO)
Learning to hop backwards

EPG
Does it actually behave like a loss function?

- Decoupled from hypothesis space, optimizer, etc?
- GD minimizes the actual objective and converges?
- General purpose — reusable for multiple tasks?
Decoupled from hypothesis space, optimizer, etc?

**Meta-train**

2 layers of 64 tanh units

**Meta-test**

2 layers of 256 tanh units  
2 layers of 64 ReLU units  
4 layers of 64 tanh units
GD minimizes the actual objective and converges?

A is meta-trained on fixed number of steps
B is meta-trained on variable number of steps
Does the loss function generalize to a **out of distribution** task?
Prior knowledge is an loss function

Prior knowledge is a recurrent policy

EPG

RL$^2$ [Duan et al., 2016]
(b) Right direction (as metatrained)   (c) Left direction (generalization)
Learning surrogate objectives

\[ \phi^* = \arg \min_{\phi} O(\arg \min_{\theta} L_{\phi}(\theta)) \]

O is hard to optimize (e.g., RL, GANs, energy-based model, etc)
L is easy to optimize
Cultural evolution
Back to the basics
Modular Co-evolution of Control and Morphology

Acts as single agent upon joining

Rewards are shared

Input = \textit{Local} Sensory State

Output = Torques, Link, Unlink

“Modular” Self-assembling Morphologies

\[
\max_{\theta} \sum_{i=1}^{n} \mathbb{E}_{d^i \sim \pi^i_\theta} \left[ \sum_{t} r^i_t \right]
\]

Dynamic Graph Networks

Environments

Standing Up

Loicomotion

Training
How well Vanilla RL work?
Vanilla Reinforcement Learning
Instead, we start with primitive limbs and allow them to self-assemble
What kind of knowledge generalizes?

Goal: agents with this kind of prior knowledge