

Deep Predictive Coding for Spatiotemporal Representation Learning

Marcio Fonseca

Master of Science
Cognitive Science
School of Informatics
University of Edinburgh
2018

Abstract

The recent advances and pitfalls of deep learning approaches reignited the debate about the importance of innate structures or inductive biases humans use to learn common sense with limited supervision. In machine learning parlance, common sense reasoning relates to the capacity of *learning representations* that disentangle hidden factors behind spatiotemporal sensory data. In this work, we hypothesise that the predictive coding theory of perception and learning from neuroscience literature may be a good candidate for implementing such common sense inductive biases. We build upon a previous deep learning implementation of predictive coding by Lotter et al. (2016) and extend its application to the challenging task of inferring abstract, everyday human actions such as *cooking* and *diving*. Furthermore, we propose a novel application of the same architecture to process auditory data, and find that with a simple sensory substitution trick, the predictive coding model can learn useful representations. Our transfer learning experiments also demonstrate good generalisation of learned representations on the UCF-101 action classification dataset.

Acknowledgements

I would like to offer my special thanks to all people and organisations that contributed to this research/free-energy-minimisation endeavour.

First, to my supervisor Dr Shay Cohen for all the support, precise guidance, and trust deposited in my work.

To my tutor Richard Shillcock, for the recommendation of valuable bibliographical resources and the pleasant conversations about Gestalt psychology.

To *the Cohort* research group, for the insightful discussions about NLP and Machine Learning. In particular, thanks to Nikos Papasasantopoulos for the early discussions on the viability of my research proposal and Lyu Chunchuan for the invaluable review on the mathematics of predictive coding theory.

To the Chamber of Deputies of Brazil, for the crucial support for pursuing the masters degree.

To my lovely wife Stephanie, for the unwavering support and care during this Scottish adventure.

Finally, to my family, my best source of inspiration and living example of integrity and perseverance.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Marcio Fonseca)

Table of Contents

1	Introduction	1
2	Background	4
2.1	Representation learning	4
2.1.1	Learning representations by minimising free energy	6
2.2	The free-energy principle (FEP)	8
2.3	From FEP to predictive coding	12
2.3.1	Approximating the posterior of environmental causes	12
2.3.2	A generative story of sensory states	14
2.3.3	Hierarchical processing in the brain	16
2.4	Representation learning from videos	16
3	Methods	20
3.1	A deep predictive coding model	20
3.1.1	Model building blocks	20
3.1.2	Limitations	22
3.2	Unsupervised pre-training	23
3.3	Extracting spatiotemporal representations	25
3.4	Supervised action recognition	25
3.5	Sensory substitution	26
4	Experiments	28
4.1	Next-frame predictions	28
4.2	Small-scale action recognition	31
4.3	Exploring the audio modality	33
4.4	Transfer learning	35
4.4.1	Next-frame predictions	36
4.4.2	Action recognition	36

5 Discussion and Future work	41
5.1 Scaling predictive coding training	42
5.2 Multimodal predictive coding	43
5.3 Integrating action	43
5.4 Learning intuitive physics	44
6 Conclusion	45
Bibliography	46

List of Figures

2.1	Schematic illustration of representation learning with disentangled hidden factors (red circles). Subsets of explanatory factors learned when solving task A are relevant and can improve the performance on tasks B and C. Reprinted from Bengio et al. (2013) Figure 1.	6
2.2	A neural network architecture of a Helmholtz machine. Solid and dotted lines between layers represent the bottom-up recognition and top-down generative connections respectively. s_j is the binary activity of unit j in layer J . Reprinted from Hinton et al. (1995) Figure 1.	9
2.3	The free-energy principle schematic. A self-organising system minimises the entropy of sensory states s (\tilde{s} is a generalised version introduced in Section 2.3.2) by minimising the free energy F . This is accomplished by adjusting the internal states μ (improving the recognition model) or acting on the world (sampling expected sensory data). Reprinted from Friston (2010) Box 1.	10
2.4	Schematic of message passing in a hierarchical predictive coding model. Red arrows indicate bottom-up prediction error propagation while black arrows indicate top-down activity predictions. Reprinted from Friston and Kiebel (2009) Figure 1.	17
2.5	The "shuffle and learn" unsupervised representation learning approach. Three siamese convolutional networks are trained end-to-end to classify a sequence of frames as temporally coherent or not. Reprinted from Misra et al. (2016) Figure 2.	18
3.1	The predictive coding architecture. Errors E_l are calculated between input A_l and prediction \hat{A}_l and passed to the upper layers. Reprinted from Lotter et al. (2016) Figure 1.	22

3.2	Sample frames from the Moments in Time dataset. Note that many types of entities such as animals, machines, and people can perform the <i>flying</i> action. Reprinted from Monfort et al. (2018) Figure 1. . . .	24
3.3	Action classification architecture showing two predictive coding layers. Representations R_l for each layer are concatenated after a spatial pooling operation. The resulting tensor is flattened and passed as input to an action classifier. Adapted from Lotter et al. (2016) Figure 1. . .	27
3.4	A sequence of spectrogram frames.	27
4.1	Last five frame predictions from a 10-frame timestep sequence sampled from the <i>exercising</i> class. First row shows ground truth frames. Second row shows predictions for a predictive coding model with random weights. The last two rows show predictions for models trained on KITTI dataset, and on 67 hours of videos from the Moments in Time dataset respectively. KITTI model predictions are blurrier and tend to be more similar to the previous frame.	30
4.2	Last five frame predictions from a 10-frame timestep sequence sampled from the <i>twisting</i> class. First row shows ground truth frames. Second and third rows show predictions for predictive coding models trained on KITTI dataset and trained on 67 hours of videos from the Moments in Time dataset respectively.	30
4.3	Last five spectrogram frame predictions from a 10-frame timestep sequence sampled from the <i>speaking</i> class. First row shows ground truth frames. Second and third rows show predictions for predictive coding model trained on KITTI dataset and trained on 37 hours of videos from the Moments in Time dataset respectively. Predictions for the model with random weights are omitted as they are mostly black frames. . .	35
4.4	Last five spectrogram frame predictions from a 10-frame timestep sequence sampled from the <i>CliffDiving</i> class. First row shows ground truth frames. Second and third rows show predictions for predictive coding model trained on KITTI dataset and trained on 67 hours of videos from the Moments in Time dataset respectively. The model captures the overall camera movement and the position of the diver but falls short of figuring out the finer details.	37

4.5 Last five spectrogram frame predictions from a 10-frame timestep sequence sampled from the *CliffDiving* class. First row shows ground truth frames. Second and third rows show predictions for predictive coding model trained on KITTI dataset and trained on 67 hours of videos from the Moments in Time dataset respectively. Even without camera movements, the model fails to predict the body movements and tends to copy the previous frames. 37

List of Tables

3.1	Versions of the predictive coding models used across the experiments.	24
3.2	Recurrent neural network classifier layers.	26
4.1	Average frame prediction errors (MSE) for different pre-trained models on a held-out set of 1000 videos spanning 10 action classes. Relative changes are computed in relation to the "copy last frame" baseline.	29
4.2	Classification accuracies (percentage) for different pre-trained models on a held-out set of 100 videos per class. First set of rows list results for a baseline model using features extracted from the VGG16 convolutional image classifier (Simonyan and Zisserman, 2014b).	32
4.3	Average frame prediction errors (MSE) for different pre-trained models on a held-out set of 622 videos spanning 10 action classes. Relative changes are computed in relation to the "copy last frame" baseline. . .	34
4.4	Classification accuracies (percentage) for different pre-trained models on a held-out set of 60 videos per class. First set of rows list results for a baseline model using features extracted from the VGG16 convolutional image classifier (Simonyan and Zisserman, 2014b).	36
4.5	Accuracies (top-1 percentage) for different pre-trained models on test set of UCF-101 split 1. We also include results for the CNN tuple verification (Misra et al., 2016) and an LSTM classifier trained on top an Inception convolutional network trained from scratch (Carreira and Zisserman, 2017).	38
4.6	Accuracies (top-1 percentage) for different pre-trained models on test set of UCF-101 split 1 (only videos from the 51 classes that contain audio).	39

4.7	Accuracies (top-1 percentage) for models trained using representations from different layers of a PredNet pre-trained on 67 hours of visual data. PredNet Video + Audio is an ensemble with the predictive coding model pre-trained on 37 hours of auditory data. Results are reported for the test set of UCF-101 split 1.	40
-----	---	----

Chapter 1

Introduction

Learning common sense from the world with a limited amount of data and supervision is a remarkable capability of human cognition. Notably, infants in their second year of life experience an impressive increase in the vocabulary acquisition known as vocabulary explosion (McMurray, 2007). This cognitive phenomenon builds upon several previous cognitive milestones, such as understanding object permanence (Baillargeon et al., 1985), spatiotemporal continuity (Spelke et al., 1994), inertia, gravity, and other knowledge about world dynamics that require extensive observation and interaction with the environment, and are essential building blocks of human-like intelligence.

In contrast, recent success in Artificial Intelligence is mostly circumscribed to probabilistic models that capture patterns from massive amounts of static, human-curated datasets. Most of the time, these models learn to infer from a probabilistic distribution over symbolic entities (e.g., language sentences) or raw sensory data (e.g., images and sounds) to a set of high-level symbolic categories relevant to a particular problem. As a result, the performance of these models is bounded by the size and quality of the datasets and, crucially, they exhibit insufficient capacity of generalising what they learn to solve novel tasks. In particular, distinguishing everyday events such as walking, running, and exercising is an open problem in computer vision research (Carreira and Zisserman, 2017; Monfort et al., 2018).

In fact, current AI approaches lack the innate machinery infants use to make sense of environmental dynamics. Such inductive biases should exploit known regularities of the world such as the constancy of the laws of physics, as proposed by Srivastava et al. (2015). Also, an intelligent agent should be able to reason about the temporal order of events, which can serve as an unsupervised learning signal for spatiotemporal data (Misra et al., 2016). In this work, we are interested in a more general, neuroscience-

inspired inductive bias in which the brain is portrayed as a hierarchical machine that improves its internal model of the world by processing the error between predicted and actual sensory stimuli. This *predictive coding* model of the human brain has been applied in theoretical neuroscience to explain processing in the visual cortex (Rao and Ballard, 1999), perceptual categorisation (Friston and Kiebel, 2009), and in cognitive science as a unified account of perception and action (Clark, 2013).

To investigate the design of machines that acquire common sense by observing the world, we capitalise on a deep learning implementation of the predictive coding model published by Lotter et al. (2016). Their deep predictive coding network was shown to learn representations that disentangle latent variables correlated to the movement of objects in synthetic and natural images. We extend their study to address the following questions:

- Can unsupervised predictive coding models learn higher-level spatiotemporal concepts, namely quotidian activities such as *driving* or *exercising*?
- Are predictive coding inductive biases general enough so that these models can also learn from auditory information?
- What are the limitations of the deep predictive coding implementation with respect to the original neuroscience theory proposed by Friston and Kiebel (2009); Rao and Ballard (1999)?

This work explores unsupervised learning from spatiotemporal data and uses video understanding tasks as a proxy to evaluate the quality of learned representations. We focus on models that can learn from large amounts of unlabelled videos and use this experience to solve downstream tasks involving smaller labelled datasets. Therefore, we *do not* pursue the solution of the action recognition problem itself, for which all the state-of-art approaches depend on a copious amount of labelled data for pre-training and often the combination of handcrafted features to encode temporal patterns (Carreira and Zisserman, 2017). Our main contributions are summarised as follows:

- Based on a theoretical review of the free-energy principle (Friston, 2010), we analyse some of the architectural limitations of Lotter et al. (2016) deep learning implementation, in particular, regarding the inference of hidden causes via free energy minimisation.
- We extend the work of Lotter et al. (2016) by using predictive coding representations to decode higher-level concepts that require the understanding of world

dynamics. The learned representations are evaluated on small-scale tasks and on UCF-101 (Soomro et al., 2012), a popular action recognition benchmark.

- We train the predictive coding model on a dataset about 60 times larger than the one used in previous work (Lotter et al., 2016) and show that model continues to improve future frame predictions, even when the training dataset includes a large number of unrelated classes.
- Inspired by sensory substitution literature from neuroscience (Stiles and Shimojo, 2015), a novel application of the predictive coding model is proposed for unsupervised representation learning from audio data. Our results suggest that the different modalities provide complementary information that is useful for the action classification task.

The rest of this work is structured as follows. In chapter 2 we succinctly introduce the problem of representation learning and show that it shares the same free-energy minimisation objective as the free-energy principle of self-organisation proposed by Friston (2010). We then review how the free-energy principle entails the predictive coding model of the brain under approximation assumptions. This review helps to identify some limitations of Lotter et al. (2016) neural network model in approximating the complex dynamical hierarchical processing proposed by Friston and Kiebel (2009). In chapter 3, we describe the methods we use to train predictive coding models, extract spatiotemporal representations, and use them to train downstream action recognition models. Chapter 4 details the experiments used to quantitatively and qualitatively evaluate the usefulness of predictive coding representations. In chapter 5, we discuss the primary experimental results and propose future research directions. Finally, chapter 6 presents a recapitulation of our findings and concluding remarks.

Chapter 2

Background

Our main motivation is to research machine learning models capable of understanding the world with limited supervision. Such models need to learn transformations from low-level input data to higher-level representations that disentangle explanatory factors for the data distribution. In other words, representations capture "general priors about the world" (Bengio et al., 2013), which are not specific to any task, including the hierarchical organisation of concepts, the manifold hypothesis (Narayanan and Mitter, 2010), and spatiotemporal coherence.

In this chapter, we introduce the representation learning problem with a focus on the main properties learned representations should exhibit to solve artificial intelligence tasks. We show that learning economic representations entails minimising an information-theoretic free energy, which is also the basis for the more general free-energy principle that unifies learning, perception, and action (Friston, 2010; Clark, 2013). We then review the approximations that lead to a theory of perception based on the processing of prediction errors, known as predictive coding (Rao and Ballard, 1999). Finally, we briefly review previous work related to our perceptual problem of interest: learning representations from videos.

2.1 Representation learning

In machine learning research, innovations in *representation learning* approaches based on deep learning are superseding traditional feature engineering methods in fields such as speech processing and computer vision (Goodfellow et al., 2016). In general, good representations capture discriminative information that facilitates other learning tasks such as classifiers and predictors. More formally, some desirable properties for rep-

representations include expressiveness, depth, abstraction, and invariance (Bengio et al., 2013). Expressiveness relates to the idea that representations should be distributed, meaning that a limited number of hidden parameters can represent high-dimensional inputs. In natural language processing, the use of word embeddings is an example of an important application of models that transform one-hot word vectors into compact distributed representations that capture semantic information (Mikolov et al., 2013).

Most of the successful representation learning approaches are also deep, in the sense that many layers of transformation are applied to generate increasingly abstract features. The positive consequence of depth is that the possible combinations of features increase exponentially as more layers are used, which according to Bengio et al. (2013), is "at the heart of the theoretical advantages behind deep learning". As an illustration, consider a traditional N-gram approach for language modelling in which the representation dimensionality increases exponentially as we increase the Markov assumption order (Jurafsky and Martin, 2014) and a recurrent neural network language model that can represent sentences of arbitrary length using a reasonably low-dimensional distributed vector (Bengio et al., 2003).

Another advantageous byproduct of deep representations is a higher abstraction, meaning that models learn to compose lower-level percepts to generate higher-level representations that are invariant to local changes in input space (Bengio et al., 2013). For instance, different layers of convolutional neural networks were shown to capture varying levels of abstraction, ranging from edges/colours at the lower layers to textures and class-specific patterns such as dogs faces and birds legs at the upper layers (Zeiler and Fergus, 2014). Although this abstraction property is mostly an empirical finding, there is ongoing research towards formalising the effects of depth on neural network feature invariance in mathematical terms. (Wiatowski and Bölcskei, 2018).

An important corollary of the above representation properties is that learned representations should be transferable to novel tasks because different subsets of the latent factors captured by abstract representations are likely to be useful for various downstream tasks (Figure 2.1). In fact, many of the state-of-the-art image and video understanding models take advantage of transfer learning from pre-trained convolutional neural networks (Carreira and Zisserman, 2017), which can be viewed as regularisers for the classification tasks (Goodfellow et al., 2016).

Our work relies on all these properties of useful representations. We assume that the predictive coding approach can learn to compose lower-level sensory data to form more abstract spatiotemporal representations that are invariant to local changes of the

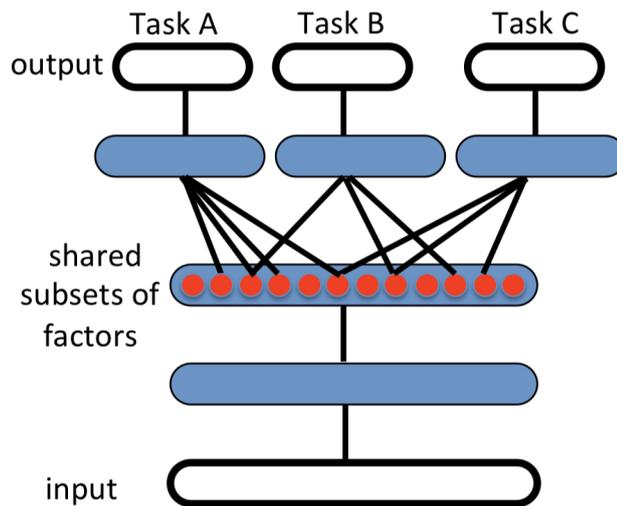


Figure 2.1: Schematic illustration of representation learning with disentangled hidden factors (red circles). Subsets of explanatory factors learned when solving task A are relevant and can improve the performance on tasks B and C. Reprinted from Bengio et al. (2013) Figure 1.

input data. In our case, learned representations should exhibit some degree of invariance when the model is exposed to videos of portraying a *running dog* and a *running person*, which allows a downstream model to discriminate the concept *running*. If our hypothesis is correct, then we should be able to use the learned representations to solve video classification tasks reasonably.

2.1.1 Learning representations by minimising free energy

The outline of an unsupervised representation learning application is straightforward. Given an unlabelled dataset X , a feature learner \mathcal{L} returns an encoder f that maps the raw input data to a representation in a lower-dimensional space. This unsupervised pre-training step is usually followed by an optional fine-tuning procedure, in which a learner \mathcal{T} uses the pre-trained function f , the data X , and the task-specific labels Y to output a new tuned function f . This strategy is described in Algorithm 1 and is an important component of our experimental methods detailed in Chapter 3.

Once the general unsupervised pre-training approach is defined, the design of the learner \mathcal{L} determines the capacity of capturing the world priors mentioned above. Bengio et al. (2007) proposed a *greedy layer-wise* unsupervised pre-training procedure that provided better initialisation parameters for training deep neural networks. In this pre-

Algorithm 1 A general unsupervised pre-training algorithm. The feature learner \mathcal{L} returns an encoder f that maps raw input data to a lower-dimensional representation. Adapted from Goodfellow et al. (2016) Algorithm 15.1.

Require: X, Y

- 1: $\hat{X} = X$ ▷ preserve the original data
 - 2: $f \leftarrow \mathcal{L}(\hat{X})$ ▷ unsupervised pre-training
 - 3: **if** fine-tune **then**
 - 4: $f \leftarrow \mathcal{T}(f, X, Y)$
 - 5: **return** f
-

training mechanism, each layer uses the features from the lower layers as data for unsupervised training and then is stacked to the previous layers to form a deep model.

Interestingly, more than one decade before the greedy layer-wise model publication, Hinton et al. (1995) devised another kind of unsupervised training procedure dubbed *wake-sleep* algorithm. This approach introduced the concept of bottom-up "recognition" connections that transform inputs into hidden representations and top-down generative connections that reconstruct the representations of the lower layers. During the wake phase, the generative model weights are adjusted to better approximate the representations generated by the bottom-up connections. Conversely, in the sleep phase, the recognition network is trained to predict the "fantasies" produced by the top-down generative model.

In this alternation between recognition and generative connections, the wake-sleep algorithm minimises the average number of bits required to "exchange messages" between adjacent layers. In other words, there is an underlying objective to reduce the entropy of the "total representation" of a given input. As a consequence, the cost function to be minimised in the wake phase is the sum of the description lengths of the hidden states α and the description lengths of the input vector d given the hidden states, defined in the following way (Hinton et al., 1995):

$$C(\alpha, d) = C(\alpha) + C(d | \alpha) \tag{2.1}$$

$$= \sum_{l \in L} \sum_{j \in l} C(s_j^\alpha) + \sum_i C(s_i^d | \alpha), \tag{2.2}$$

where the description length is $C(s_j^\alpha)$ is the description length of a unit j within the total representation α . Assuming all units are binary, the description length is defined

in terms of the unit activations s_j^α and the probabilities p_j^α as follows:

$$C(s_j^\alpha) = -s_j^\alpha \ln p_j^\alpha - (1 - s_j^\alpha) \ln(1 - p_j^\alpha). \quad (2.3)$$

The binary probability distributions $(p_j^\alpha, 1 - p_j^\alpha)$ for each unit j are calculated using the top-down generative connection weights w_{kj} from a upper layer units k to the unit j as follows:

$$P(s_j = 1) = \frac{1}{1 + \exp(-b_j - \sum_k s_k w_{kj})}, \quad (2.4)$$

where b_j is the unit bias. We then can take advantage of the entropy of the recognition distribution and define a final cost function that considers the many alternative descriptions α of an input vector (Hinton and Zemel, 1994):

$$C(d) = \sum_{\alpha} Q(\alpha | d) C(\alpha, d) - \left(- \sum_{\alpha} Q(\alpha | d) \ln Q(\alpha | d) \right). \quad (2.5)$$

As Hinton et al. (1995) noted, there is an insightful analogy between Equation 2.5 and the Helmholtz free energy concept from thermodynamics (McNaught and McNaught, 1997), with the first term $\sum_{\alpha} Q(\alpha | d) C(\alpha, d)$ corresponding to the internal energy of the system. Thus, the cost in Equation 2.5 is minimised when the distribution $Q(\alpha | d)$ is the Boltzmann distribution (Sharp and Matschinsky, 2015):

$$P(\alpha | d) = \frac{\exp(-C(\alpha, d))}{\sum_{\beta} \exp(-C(\beta, d))}. \quad (2.6)$$

Since the exact calculation of this distribution is intractable, Hinton et al. (1995) devised a way to make $Q(\alpha | d)$ approximate the optimal distribution. In the sleep phase, each layer is trained to predict the hidden layer sampled from the generator, a strategy that was demonstrated to work well empirically (Hinton et al., 1995).

The architecture described above, dubbed Helmholtz machine (Dayan et al., 1995), is illustrated in Figure 2.2. Despite its simplicity, this representation learning approach includes several relevant elements such as hierarchical layer-wise training, generative and recognition models, and the crucial inductive bias that encourages the minimisation of the free energy in Equation 2.5. In the next sections, we review how these building blocks are combined and extended to build models of perception, action, and learning in the brain.

2.2 The free-energy principle (FEP)

In the wake-sleep algorithm, the objective of reducing the information-theoretic description length of the data led to a solution involving a generative model that min-

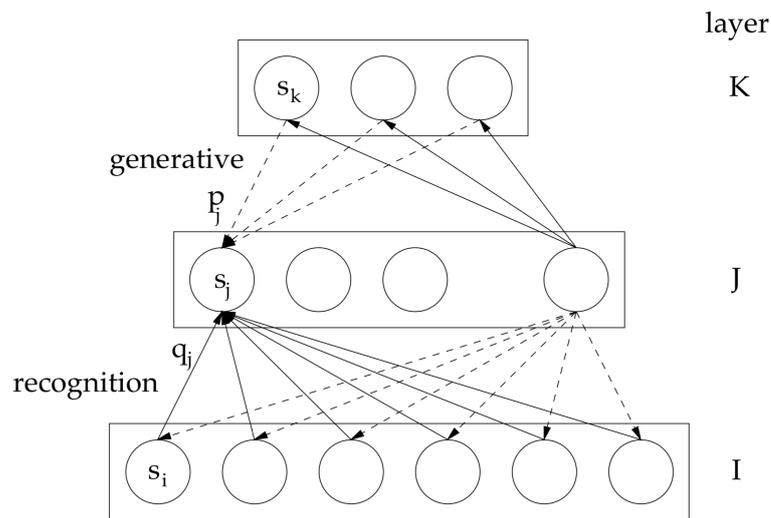


Figure 2.2: A neural network architecture of a Helmholtz machine. Solid and dotted lines between layers represent the bottom-up recognition and top-down generative connections respectively. s_j is the binary activity of unit j in layer J . Reprinted from Hinton et al. (1995) Figure 1.

imises a Helmholtz free energy (see Section 2.1.1). In this section, we approach this problem from a top-down perspective by posing a simple, yet tricky question: *what distinguish living from non-living entities?* According to Friston (2010), the answer lies in the intrinsic motivation that self-organising systems at any scale such as cells, brains, and societies have to resist a natural tendency to disorder. Living systems achieve this goal by minimising free energy using its internal model of the world, which by the free-energy principle, is equivalent to maximising the evidence of its own existence (Friston, 2010).

To understand the mathematical formulation of the free-energy principle, we consider a hypothetical self-organising system portrayed in Figure 2.3. Our first important concern is to define a criterion to separate the system from the rest of the environment, that is, to define which states are internal (μ) and external (x) to the living system. The Markov blanket concept (Barber, 2012; Clark, 2017) provides such formalism by defining, within a complex set of interacting variables, a set of nodes (the blanket) that make a target node conditionally independent of all other variables in the system.

To resist disorder, the self-organising system has to keep its internal variables within a limited set of possible states. For instance, suppose the agent needs to maintain its physiological states such as the body temperature within a narrow range that allows its metabolism to function properly. From an information-theoretic perspective,

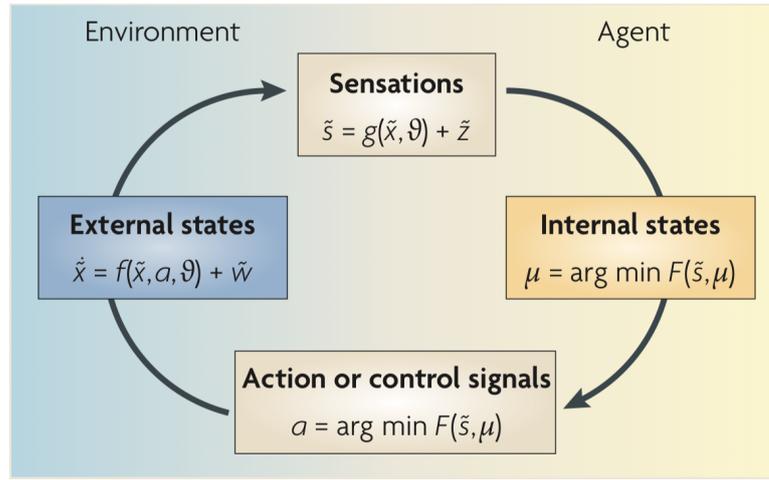


Figure 2.3: The free-energy principle schematic. A self-organising system minimises the entropy of sensory states s (\tilde{s} is a generalised version introduced in Section 2.3.2) by minimising the free energy F . This is accomplished by adjusting the internal states μ (improving the recognition model) or acting on the world (sampling expected sensory data). Reprinted from Friston (2010) Box 1.

this homeostatic control corresponds to reducing the long-term entropy or the uncertainty of the nodes in the Markov blanket. As will be shown in this section, the free energy is an upper bound to long-term average surprisal (Clark, 2017), and therefore, minimising free energy is a proxy for reducing the uncertainty of the system states.

Formally, the surprisal quantity mentioned above is define as follows:

$$I(s) = -\ln p(s), \quad (2.7)$$

where $p(s)$ is the probability of perceiving the sensory state s . These sensory states are explained by hidden causes ϑ outside the Markov blanket, and thus, must be inferred via Bayesian inference. Under the free-energy principle, the agent has to build an internal probabilistic model $p(s, \vartheta)$ that captures the joint distributions of the hidden environmental states and the perceived sensory inputs, and can be factorised as follows:

$$p(s, \vartheta) = p(s | \vartheta)p(\vartheta), \quad (2.8)$$

where $p(s | \vartheta)$ is the likelihood or noise model that captures assumptions about how external causes are mapped to sensory states and $p(\vartheta)$ captures priors beliefs about the causes. Ideally, the posterior probability of the hidden causes would be calculated using Bayes rule (Buckley et al., 2017):

$$p(\vartheta | s) = \frac{p(s | \vartheta)p(\vartheta)}{\int p(s | \vartheta)p(\vartheta)d\vartheta}. \quad (2.9)$$

However, the calculation of the partition function in the denominator is often intractable. To sidestep this issue, we choose an approximate recognition distribution $q(\vartheta | \mu)$ and calculate the model log evidence $\ln p(s)$ as follows:

$$\ln p(s) = \ln \int p(s | \vartheta) p(\vartheta) d\vartheta \quad (2.10)$$

$$= \ln \int p(s, \vartheta) d\vartheta \quad (2.11)$$

$$= \ln \int \frac{q(\vartheta | \mu)}{q(\vartheta | \mu)} p(s, \vartheta) d\vartheta \quad (2.12)$$

$$= \ln E_q \left[\frac{p(s, \vartheta)}{q(\vartheta | \mu)} \right] \quad (2.13)$$

$$\geq E_q [\ln p(s, \vartheta)] - E_q [\ln q(\vartheta | \mu)], \quad (2.14)$$

where the last step follows from the Jensen's inequality (Barber, 2012). This evidence lower bound on the right hand side is calculated as the energy term $E_q \ln [p(s, \vartheta)]$ minus the entropy $E_q [\ln q(\vartheta)]$, and is similar to the Helmholtz energy in Equation 2.5. Finally, we can define the variational free energy F as the negative of this evidence lower bound:

$$F = -E_q [\ln p(s, \vartheta)] + E_q [\ln q(\vartheta | \mu)] \quad (2.15)$$

$$\geq -\ln p(s). \quad (2.16)$$

Thus, equation 2.15 shows formally that minimising the free energy F is an indirect way to minimise sensory surprise (Friston, 2010) and also, that the self-organising system must implement a generative model $p(s, \vartheta)$ that relates causes to sensory data. Alternatively, we can write the Equation 2.15 in terms the the Kullback-Leibler divergence:

$$F = D_{KL}(q(\vartheta | \mu) || p(\vartheta | s)) - \ln p(s). \quad (2.17)$$

Since the second term does not depend on $q(\vartheta)$ and the Kullback-Leibler divergence is always non-negative, we conclude that minimising free energy also implies that the recognition distribution $q(\vartheta)$ is a better approximation of the true posterior $p(\vartheta | s)$. Therefore, Equation 2.17 gives a *perceptual* view of the free-energy principle.

An important consequence of the free-energy principle is that it also accounts for action, as expressed by a third formulation of the free energy:

$$F = D_{KL}(q(\vartheta | \mu) || p(\vartheta)) - E_q [\ln p(s(a) | \vartheta)], \quad (2.18)$$

where the first term is the Bayesian surprise, that is, the divergence between the prior beliefs on the hidden causes and the approximate posterior, while the second term is the accuracy or "the surprise about sensations" (Friston, 2010). The interpretation of this equation is that the agent can act to sample the sensory information that confirms its own expectations, which is known as active inference (Friston et al., 2017).

To conclude, the free-energy principle states that a self-organising system that resists disorder must rely on a generative model of sensory states and a recognition model that approximates the posterior distribution of hidden sensory causes (Equation 2.15). The agent can maximise the evidence of its own existence by minimising a variational free energy either by changing its internal states to improve the recognition model as follows:

$$\mu = \operatorname{argmin} F(s, \mu), \quad (2.19)$$

or acting to sample new sensory information according to its expectations (Friston, 2010):

$$a = \operatorname{argmin} F(s(a), \mu). \quad (2.20)$$

2.3 From FEP to predictive coding

The free-energy principle provides a mathematical formalism for explaining how a self-organising system can resist disorder without having access to the true posterior distribution for the causes of the sensory states. However, it does not make any assumptions on how the recognition and generative models are actually implemented in a self-organising system such as the brain. In this section, we briefly present the approximation assumptions that reduce the free energy minimisation problem to a minimisation of errors between predictions and sensory input. This model is known as predictive coding and is a well-known framework for perceptual processing in the brain (Rao and Ballard, 1999).

2.3.1 Approximating the posterior of environmental causes

The first simplifying assumption defines the form of the recognition function $q(\vartheta)$ as a Gaussian, which is called the Laplace approximation (Friston, 2010). Additionally, the internal states μ corresponding to the neuronal activations are treated as a sufficient

statistics and parameterise the following Gaussian distribution (Buckley et al., 2017):

$$q(\vartheta) \equiv \mathcal{N}(\vartheta; \mu, \zeta) = \frac{1}{\sqrt{2\pi\zeta}} \exp \left\{ -(\vartheta - \mu)^2 / (2\zeta) \right\}, \quad (2.21)$$

where μ and ζ are the mean and variance respectively. Substituting this recognition density form into Equation 2.15 gives a simplified form for the free energy:

$$F = -\ln \sqrt{2\pi\zeta} - \int d\vartheta q(\vartheta) (\vartheta - \mu)^2 / (2\zeta) + \int d\vartheta q(\vartheta) E(\vartheta, s) \quad (2.22)$$

where $E(\vartheta, s) \equiv -\ln p(\vartheta, s)$ is an energy by analogy with the Helmholtz free energy (McNaught and McNaught, 1997) and s is the same sensory state from Section 2.2. The second term in the expression can be further simplified as the integral correspond to the second central moment, that is, the variance ζ (Barber, 2012), resulting in the following expression:

$$F = -\ln \sqrt{2\pi\zeta} - \frac{1}{2} + \int d\vartheta q(\vartheta) E(\vartheta, s). \quad (2.23)$$

The last term of Equation 2.19 includes the unspecified energy $E(\vartheta, s)$ that depends on hidden environmental states ϑ . To solve this issue, two further simplifications are introduced. First, the Gaussian recognition density is assumed to be "sharply peaked" around the mean μ . Second, we assume the energy $E(\vartheta, s)$ function is smooth with respect to ϑ . As a consequence, the integral in the third term would be almost zero anywhere except around the mean. Buckley et al. (2017) use these two assumptions to approximate the energy around the mean using the Taylor expansion (Barber, 2012), which simplifies the integral to:

$$\begin{aligned} \int d\vartheta q(\vartheta) E(\vartheta, s) &\approx E(\mu, s) + \left[\frac{\partial E}{\partial \vartheta} \right]_{\mu} \int d\vartheta q(\vartheta) (\vartheta - \mu) \\ &\quad + \frac{1}{2} \left[\frac{d^2 E}{d\vartheta^2} \right]_{\mu} \int d\vartheta q(\vartheta) (\vartheta - \mu)^2 \\ &= E(\mu, s) + \frac{1}{2} \left[\frac{d^2 E}{d\vartheta^2} \right]_{\mu} \zeta, \end{aligned} \quad (2.24)$$

where the last step follows from the fact that the first integral is zero (because the mean of $q(\vartheta)$ is μ) and the second integral is the variance of $q(\vartheta)$. The energy $E(\mu, s)$ is now parameterised in terms of the sufficient statistics μ in place of the causes ϑ , and for this reason is called the Laplace-encoded energy (Buckley et al., 2017). The free energy from Equation 2.23 can be rewritten as:

$$F(\mu, \zeta, s) = E(\mu, s) + \frac{1}{2} \left(\left[\frac{d^2 E}{d\vartheta^2} \right]_{\mu} \zeta - \ln 2\pi\zeta - 1 \right). \quad (2.25)$$

To eliminate the dependency on the parameter ζ , we can write the free energy in terms of a "optimal variance" (obtained by equating the derivative with respect to ζ to zero):

$$F(\mu, s) = E(\mu, s) - \frac{1}{2} \ln \{2\pi\zeta^*\}, \quad (2.26)$$

where $\zeta^* = \left[\frac{d^2 E}{d\vartheta^2} \right]_{\mu}^{-1}$. Finally, according to Friston and Kiebel (2009) this expression can be further approximated to:

$$F(\mu, s) = E(\mu, s) = -\ln p(\mu, s). \quad (2.27)$$

Therefore, the Laplace approximation of the recognition density $q(\vartheta)$ entails a free energy expression that depends only on the internal states μ and the sensory states s as opposed to the external causes ϑ (Equation 2.15) that are not directly accessible to the brain. Crucially, the brain wetware does not need to represent all the probability distribution details but only the sufficient statistics μ encoded by internal states.

2.3.2 A generative story of sensory states

From the last section, we concluded that to minimise free energy (Equation 2.27), the brain must necessarily implement the Laplace-encoded energy or the approximate generative density $-\ln p(\mu, s)$. Thus, we have to construct a generative story of how the sensory states s are generated. According to (Friston and Kiebel, 2009), this can be accomplished using "hierarchical dynamic models" as follows:

$$s = g(\mu, \theta) + z \quad (2.28)$$

$$\frac{d\mu}{dt} = f(\mu, \theta) + w, \quad (2.29)$$

where f and g are non-linear functions parameterised by θ , and z and w introduce stochastic noise with well-defined covariances. Equation 2.29 is a Langevin stochastic differential equation (Zwanzig, 1973). Friston and Kiebel (2009) also use the concept of "generalised coordinates of motion", which adds hierarchical dynamics to the model. As a consequence, the Equations 2.28 and 2.29 can be generalised as follows:

$$\tilde{s} = \tilde{g} + \tilde{z} \quad (2.30)$$

$$D\tilde{\mu} = \tilde{f} + \tilde{w}, \quad (2.31)$$

where the $\tilde{g} = [g, g', g'', \dots]^T$ is an infinite dimensional motion vector containing derivatives of the function g with respect to time. The same generalisation applies to \tilde{f} , \tilde{s} , $\tilde{\mu}$,

\tilde{z} , and \tilde{w} . The matrix D is a derivative operator that generalises the Langevin equation so that

$$\tilde{\mu}' \equiv D\tilde{\mu} = \frac{d}{dt}(\mu, \mu', \mu'', \dots). \quad (2.32)$$

The link between these generative equations and the approximate generative density lies in the idea that the generalised function \tilde{g} (Equation 2.30) describes how sensory states are inferred from internal states, that is, the conditional probability $p(\tilde{s} | \tilde{\mu})$. Similarly, Equation 2.31 describes the transitions across adjacent dynamical orders of internal states, corresponding to $p(\tilde{\mu})$. As previously defined, these two distributions are sufficient to derive the Laplace-encoded model $E(\tilde{\mu}, \tilde{s})$.

Assuming the fluctuations $\tilde{z} = \tilde{s} - \tilde{g}$ are Gaussian distributed with mean \tilde{g} and variance $\sigma_{\tilde{z}}$, we have

$$\begin{aligned} p(\tilde{s} | \tilde{\mu}) &= \prod_{n=0}^{\infty} p(s_{[n]} | \mu_{[n]}) \\ &= \prod_{n=0}^{\infty} \frac{1}{\sqrt{2\pi\sigma_{z[n]}}} \exp \left[- \{s_{[n]} - g_{[n]}\}^2 / (2\sigma_{z[n]}) \right], \end{aligned} \quad (2.33)$$

where $s_{[n]}$, $\mu_{[n]}$, $g_{[n]}$, and $z_{[n]}$ are the n -th order derivatives of s , μ , g , and z respectively. By the same token, if the fluctuations $\tilde{w} = D\tilde{\mu} - \tilde{f}$ are Gaussian distributed with mean \tilde{f} and variance $\sigma_{\tilde{w}}$, we have

$$\begin{aligned} p(\tilde{\mu}) &= \prod_{n=0}^{\infty} p(\mu_{[n+1]} | \mu_{[n]}) \\ &= \frac{1}{\sqrt{2\pi\sigma_{w[n]}}} \exp \left[- \{\mu_{[n+1]} - f_{[n]}\}^2 / (2\sigma_{w[n]}) \right], \end{aligned} \quad (2.34)$$

where $\mu_{[n]}$, $f_{[n]}$, and $w_{[n]}$ are the n -th order derivatives of s , μ , g , and z respectively. Combining the above expressions for $p(\tilde{s} | \tilde{\mu})$ and $p(\tilde{\mu})$ we can write the Laplace-encoded energy as follows (Buckley et al., 2017):

$$\begin{aligned} E(\tilde{\mu}, \tilde{\varphi}) &= \sum_{n=0}^{\infty} \left\{ \frac{1}{2\sigma_{z[n]}} [\epsilon_{z[n]}]^2 + \frac{1}{2} \ln \sigma_{z[n]} \right\} \\ &+ \sum_{n=0}^{\infty} \left\{ \frac{1}{2\sigma_{w[n]}} [\epsilon_{w[n]}]^2 + \frac{1}{2} \ln \sigma_{w[n]} \right\}, \end{aligned} \quad (2.35)$$

where $\epsilon_{z[n]} \equiv s_{[n]} - g_{[n]}$ and $\epsilon_{w[n]} \equiv \mu_{[n+1]} - f_{[n]}$. Since this energy also corresponds to the variational free energy (by the Laplace approximation), we conclude the minimising the free energy implies minimising the squared errors between predicted and actual states, which leads to the predictive coding model proposed by Rao and Ballard (1999).

2.3.3 Hierarchical processing in the brain

In the previous section, we derived the recognition and generative densities assuming the Laplace approximation. The resulting model has a dynamical structure in the sense that a hierarchy between adjacent time derivatives of states is imposed. As a further improvement, Friston and Kiebel (2009) propose the addition of a spatial hierarchy, by stacking multiple layers of the predictive coding generative process. In this case, Friston and Kiebel (2009) split the internal state into causal states μ^v that serve as link between levels and hidden states μ^x that provide memory to the system by linking the states over time. The resulting hierarchical model with M layers has the following form:

$$\tilde{\mu}^{(i)v} = \tilde{g}^{(i+1)}(\tilde{\mu}^{(i+1)x}, \tilde{\mu}^{(i+1)v}) + \tilde{z}^{(i)}, \quad i = 0, 1, \dots, M \quad (2.36)$$

$$D\tilde{\mu}^{(i)x} = \tilde{f}^{(i)}(\tilde{\mu}^{(i)x}, \tilde{\mu}^{(i)v}) + \tilde{w}^{(i)}, \quad i = 1, 2, \dots, M, \quad (2.37)$$

where $\tilde{\mu}^{(0)v}$ denotes the sensory data at the lowest level (generalising \tilde{s} in the non-hierarchical model). Finally, this expression can be rewritten in terms of state and error units as follows (Friston and Kiebel, 2009):

$$\dot{\tilde{\mu}}^{(i)v} = D\tilde{\mu}^{(i)v} - \tilde{\epsilon}_v^{(i)T} \xi^{(i)} - \xi^{(i+1)v} \quad (2.38)$$

$$\dot{\tilde{\mu}}^{(i)x} = D\tilde{\mu}^{(i)x} - \tilde{\epsilon}_x^{(i)T} \xi^{(i)} \quad (2.39)$$

$$\xi^{(i)v} = \tilde{\mu}^{(i-1)v} - \tilde{g}(\tilde{\mu}^{(i)}) - \Lambda^{(i)z} \xi^{(i)v} \quad (2.40)$$

$$\xi^{(i)x} = D\tilde{\mu}^{(i)x} - \tilde{f}(\tilde{\mu}^{(i)}) - \Lambda^{(i)w} \xi^{(i)x}, \quad (2.41)$$

where $\tilde{\epsilon}_v^{(i)}$ and $\tilde{\epsilon}_x^{(i)}$ are prediction errors. Λ and ξ modulate the prediction errors using the noise inverse variance (for full treatment refer to Friston and Kiebel (2009)). The important insight from these equations is that they define the wiring of the dynamic hierarchical system. Error units $\xi^{(i)x}$ encode intra-layer prediction errors while units $\xi^{(i)v}$ capture top-down prediction errors. The state units $\tilde{\mu}^{(i)x}$ and $\tilde{\mu}^{(i)v}$ integrate intra and lower layer activities and are responsible for the inference of hidden causes. An illustration of the architecture is shown in Figure 2.4. We refer to these equations when introducing the deep learning predictive coding implementation in Section 3.1.

2.4 Representation learning from videos

Recent research in video understanding tasks builds upon the impressive advances in image classification tasks (Krizhevsky et al., 2012). This fact is demonstrated by the

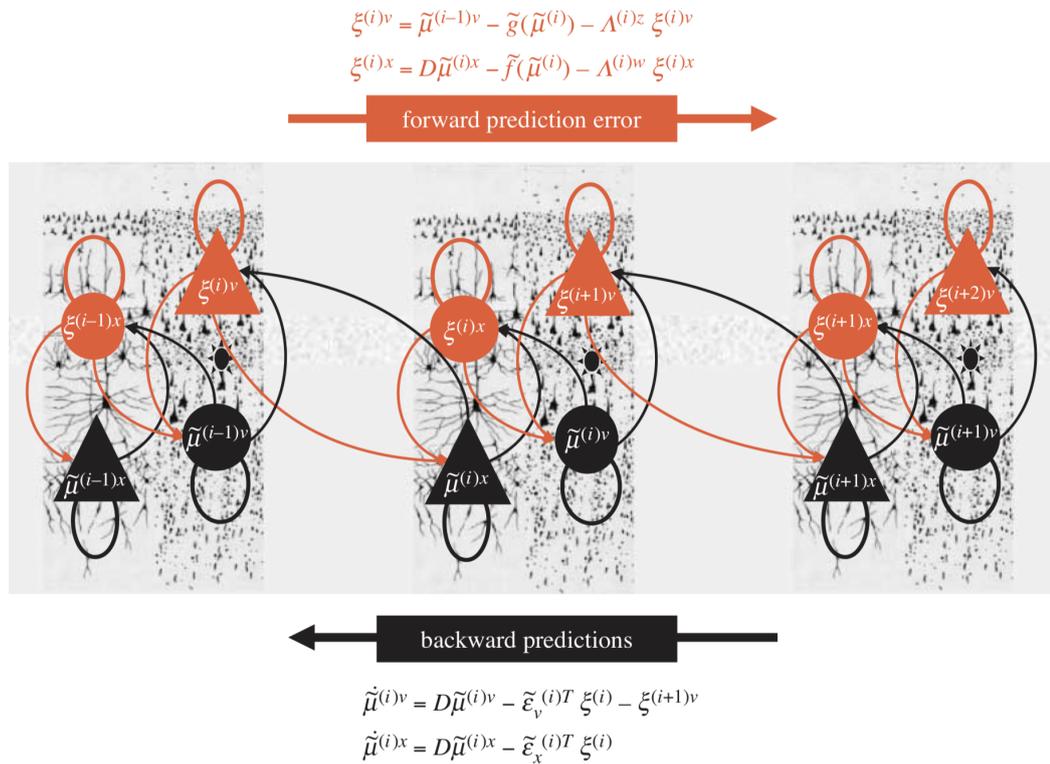


Figure 2.4: Schematic of message passing in a hierarchical predictive coding model. Red arrows indicate bottom-up prediction error propagation while black arrows indicate top-down activity predictions. Reprinted from Friston and Kiebel (2009) Figure 1.

prevalence of deep convolutional neural networks (LeCun et al., 1995) in this field (Carreira and Zisserman, 2017). Besides, the representations learned by these image classifiers were shown to improve the performance of action recognition models significantly (Simonyan and Zisserman, 2014a), serving as an effective regularisation technique.

The use of such pre-trained models comes, however, at a price. Different from the unsupervised pre-training approach we described in Section 2.1, these classification models are trained with *labelled* data that are expensive to obtain. Moreover, there are domain adaptation issues (Glorot et al., 2011), and the usefulness of the representations decreases when the downstream task data distribution differs significantly from the pre-training dataset. For this reason, there is a substantial effort in building vast labelled video datasets (Monfort et al., 2018; Carreira and Zisserman, 2017) that provide vast improvements through supervised representation learning.

Another approach to this problem is to leverage the copious amount of *unlabelled* video data that are readily available, carrying useful information about how the world

dynamics unfold. Recent work by Srivastava et al. (2015) applied recurrent encoder-decoder networks models that learn representations by predicting the future and reproducing the input sequence of frames at the same time. However, they trained the recurrent networks using percepts from models pre-trained on the ImageNet classification dataset (Deng et al., 2009), which undermines the idea of unsupervised learning.

Sharing our pure unsupervised learning motivation, (Misra et al., 2016) showed that convolutional architectures (Figure 2.5) could learn representations by solving a temporal coherence task. Their main idea is to sidestep the complexity of using generative models and use instead, a simple binary classification task that teaches the model to distinguish between coherent and shuffled frame sequences. We use Misra et al. (2016) results as baseline in our transfer learning experiment detailed in Section 4.4. One advantage of the predictive coding model we introduce in Section 3.1 over this "shuffle and learn" approach is the use of a generative model that allows the evaluation of representations even without a downstream classification task.

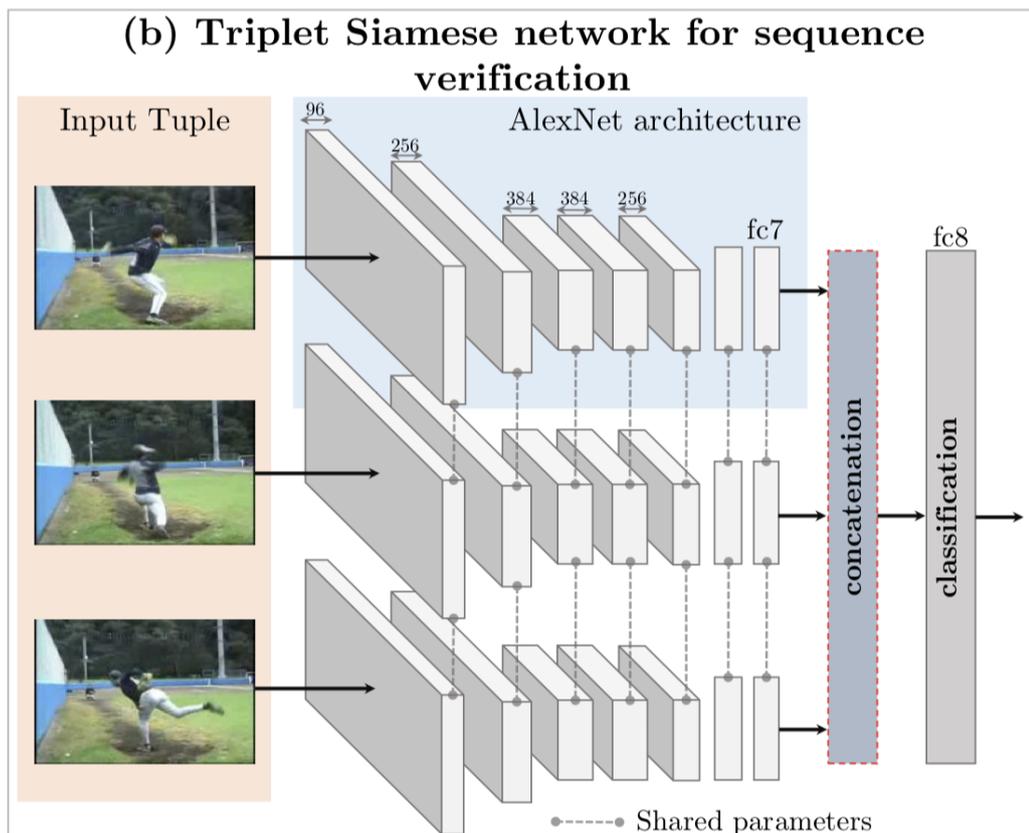


Figure 2.5: The "shuffle and learn" unsupervised representation learning approach. Three siamese convolutional networks are trained end-to-end to classify a sequence of frames as temporally coherent or not. Reprinted from Misra et al. (2016) Figure 2.

A limited research effort has been employed to leverage auditory information present in videos. Most of the solutions use specialised models for sound classification followed feature fusion (Snoek et al., 2005) or model ensembling to improve the performance of vision-only models (Monfort et al., 2018). Convolutional neural networks were also shown to perform well on classification tasks using audio spectrograms (Wang et al., 2016). In Section 4.3, we demonstrate that the same predictive coding architecture used for the visual modality can learn representations from unlabelled dynamic audio spectrograms. Finally, we suggest multimodal predictive coding research directions in Section 5.2.

Chapter 3

Methods

Our goal is to empirically determine if predictive coding networks provide useful biases to capture patterns from spatiotemporal data. In particular, learned representations should disentangle explanatory variables for the observed inputs and also be useful for downstream supervised models (Bengio et al., 2013). In this chapter, we present the methods used to test if predictive coding representations exhibit those properties. First, we describe the a machine learning implementation of the predictive coding theory proposed by Lotter et al. (2016) *vis-à-vis* with the original mathematical formalism by Friston and Kiebel (2009). Also, we detail the unsupervised pre-training step, including the datasets used to generate each model used across the experiments. Finally, we explain how representations are extracted and used by the action recognition classifiers.

3.1 A deep predictive coding model

To perform our experiments, we need a machine learning model that learns representations from high-dimensional spatiotemporal data under the free-energy principle. In this section, we describe how Lotter et al. (2016) translate the predictive coding formalism developed in Section 2.3 into neural network constructs such as convolutional and recurrent layers trained via backpropagation (LeCun et al., 1988).

3.1.1 Model building blocks

The deep predictive coding model (PredNet) (Lotter et al., 2016) shares the same overall idea of the predictive coding theory proposed by Friston and Kiebel (2009); Rao and Ballard (1999). First, it has a dynamical hierarchy, since it is implemented as a

recurrent model that process a sequence of inputs x_t . Also, it is hierarchical spatially, with the same prediction module stacked to form a multilayer architecture (Figure 3.1). Each module has the following components:

- An input convolutional layer (LeCun et al., 1995), A_l , receiving a raw sensory data x_t at lowest layer $l = 0$ or the prediction error from the lower layer when $l > 0$:

$$A_l^t = \begin{cases} x_t & \text{if } l = 0 \\ \text{MAXPOOL}(\text{RELU}(\text{CONV}(E_{l-1}^t))) & l > 0. \end{cases} \quad (3.1)$$

- A recurrent convolutional representation layer (Xingjian et al., 2015), R_l , receiving as input an intra-layer error E_l and the upper-level representation R_{l+1} :

$$R_l^t = \text{CONVLSTM}(E_l^{t-1}, R_l^{t-1}, \text{UPSAMPLE}(R_{l+1}^t)). \quad (3.2)$$

- A prediction layer, \hat{A}_l , for the next sensory data taking as input the representation activations R_l :

$$\hat{A}_l^t = \text{RELU}(\text{CONV}(R_l^t)). \quad (3.3)$$

- An error layer, E_l , calculating the difference between the prediction \hat{A}_l and the actual data A_l followed by a rectified linear unit (ReLU):

$$E_l^t = [\text{RELU}(A_l^t - \hat{A}_l^t); \text{RELU}(\hat{A}_l^t - A_l^t)], \quad (3.4)$$

where the split between positive and negative populations is motivated by the analogy with types of cells in the visual system (e.g., on-center, off-surround ganglion cells) (Lotter et al., 2016).

The model is trained via backpropagation to minimise the weighted sum of the activity of the error units as follows:

$$L_{\text{train}} = \sum_t \lambda_t \sum_l \frac{\lambda_l}{n_l} \sum_{n_l} E_l^t, \quad (3.5)$$

where E_l^t represents the error units in layer l at timestep t and n_l is the number of error units in layer l . The timestep weights λ_t and layer-wise weights λ_l are hyperparameters.

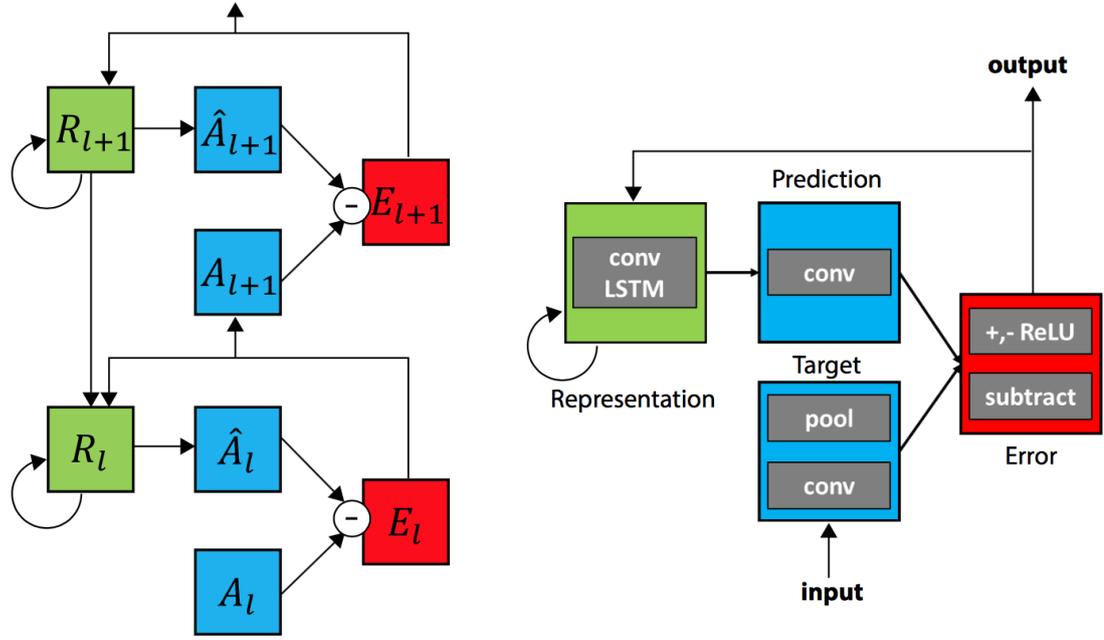


Figure 3.1: The predictive coding architecture. Errors E_l are calculated between input A_l and prediction \hat{A}_l and passed to the upper layers. Reprinted from Lotter et al. (2016) Figure 1.

3.1.2 Limitations

In this section, we explore the differences between the PredNet computation graph and the predictive coding model of the brain under the free-energy principle. As shown in Section 2.2, the free-energy principle mandates the implementation of a generative model of sensory states. In the predictive coding case, this generative story is expressed by the following equations (Friston and Kiebel, 2009):

$$\dot{\tilde{\mu}}^{(i)v} = D\tilde{\mu}^{(i)v} - \tilde{\epsilon}_v^{(i)T} \xi^{(i)} - \xi^{(i+1)v} \quad (3.6)$$

$$\dot{\tilde{\mu}}^{(i)x} = D\tilde{\mu}^{(i)x} - \tilde{\epsilon}_x^{(i)T} \xi^{(i)} \quad (3.7)$$

$$\xi^{(i)v} = \tilde{\mu}^{(i-1)v} - \tilde{g}(\tilde{\mu}^{(i)}) - \Lambda^{(i)z} \xi^{(i)v} \quad (3.8)$$

$$\xi^{(i)x} = D\tilde{\mu}^{(i)x} - \tilde{f}(\tilde{\mu}^{(i)}) - \Lambda^{(i)w} \xi^{(i)x}, \quad (3.9)$$

where \tilde{f} and \tilde{g} are non-linear functions parameterised by θ , $\tilde{\mu}$ are internal states, $\tilde{\epsilon}$ denote prediction errors and ξ are precision-weighted errors (refer to Section 2.3.3 for details). The PredNet most likely counterpart for Equation 3.6 is given by Equation 3.2, where the representation R_l^t at layer l and timestep t is related to the causal state $\dot{\tilde{\mu}}^{(i)v}$. The recurrent ConvLSTM layer has a similar role as the derivative operator D , implementing a transition in the time dimension. However, there is no explicit

implementation of the hidden state $\dot{\tilde{\mu}}^{(i)x}$ and it is not clear if the LSTM hidden state (Hochreiter and Schmidhuber, 1997) is an equivalent construct.

Concerning the error units, Equations 3.8 and 3.9 roughly correspond to the input (Equation 3.1), prediction (Equation 3.3), and error (Equation 3.4) layers, where \hat{A}_l^t and A_l^t are related to $\tilde{g}(\tilde{\mu}^{(i)})$ and $\tilde{f}(\tilde{\mu}^{(i)})$ respectively, suggesting that the non-linear linear functions \tilde{f} and \tilde{g} are implemented by the composition of a ReLU activation function and a convolutional layer. Again, there is no counterpart in this deep learning model for the hidden error unit $\xi^{(i)x}$ and, importantly, no recurrent layer in the error component E_l^t . Finally, there is no modelling of uncertainty Λ to weight prediction errors in Lotter et al. (2016) model. From this comparison, we conclude that there are aspects in the deep learning implementation that can be improved to make the model closer to the original predictive coding theory.

Beyond the architectural differences, there is a fundamental issue concerning how inference is treated in each model. In the predictive formulation by Friston and Kiebel (2009), the brain dynamics is assumed to perform inference by optimising the internal states $\tilde{\mu}$ so that free-energy is minimised. Changes in synaptic gain and efficacy are implemented by other optimisation processes that happen at slower timescales (Buckley et al., 2017). On the other hand, Lotter et al. (2016) deep learning implementation optimises the model weights only during the learning step to minimise the weighted errors as defined in Equation 3.5. Therefore, during inference, there is no formal guarantee that the neural network activations are changing to minimise a quantity related to the free energy. Further work is needed to verify the influence of this limitation on the quality of learned representations.

3.2 Unsupervised pre-training

The first part of the experiments consists in training the predictive coding architecture using an unlabelled action recognition dataset. The main idea is that the more data we use to train the model, the more "common sense" it should get about the world and, as a consequence, it should be better at solving other tasks.

As stated in our project proposal, the Moments in Time dataset (Monfort et al., 2018), which is a large-scale activity recognition dataset, is our dataset of choice for obtaining unlabelled videos. This dataset is particularly suited to this experiment because it has a broad semantic coverage of actions (339 actions/events) with each action being performed by several kinds of entities. For instance, the action *flying* can be

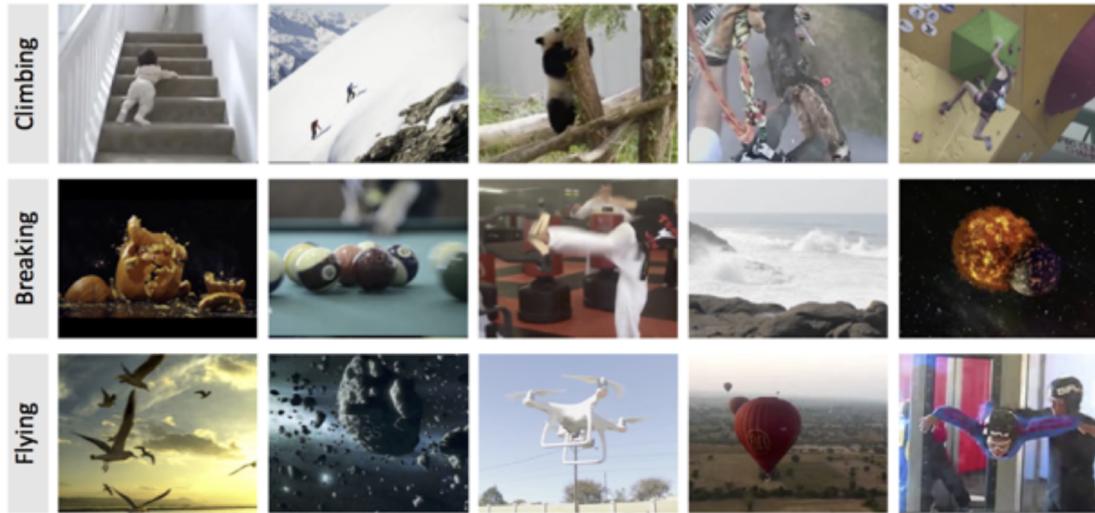


Figure 3.2: Sample frames from the Moments in Time dataset. Note that many types of entities such as animals, machines, and people can perform the *flying* action. Reprinted from Monfort et al. (2018) Figure 1.

performed by a bird, a person or a drone. This variation encourages the unsupervised learning model to separate the action dynamics from a specific kind of entity. Figure 3.2 shows sample frames for some classes.

Another relevant characteristic is that the videos have fixed duration of three seconds, which allows us to curate two balanced subsets with 3 hours (10 actions) and 67 hours (200 actions) of video data. Each of these datasets is used to train different versions of the unsupervised model, which are compared to a model with random weights (no training) and a version trained on the KITTI dataset (Geiger et al., 2013), kindly provided by Lotter et al. (2016). A summary of the different pre-trained models is shown in Table 3.1.

Model name	Dataset	Frames	Hours	Action classes
PredNet random	-	0	0	-
PredNet KITTI	KITTI	$\approx 41K$	≈ 1	-
PredNet Moments 3h	Moments in Time	$\approx 120K$	≈ 3.3	10
PredNet Moments 67h	Moments in Time	$\approx 2.4M$	≈ 66.6	200

Table 3.1: Versions of the predictive coding models used across the experiments.

As explained in Section 3.1, the predictive coding model is trained in an unsupervised way to predict the next frame using a top-down generative model. The er-

rors between predictions and the actual frames are propagated bottom-up to update the prior for new predictions. We follow the same training configuration used in the original PredNet implementation proposed by Lotter et al. (2016), with four modules consisting of 3x3 convolutional layers with 3, 48, 96, and 192 filters. The videos are subsampled at ten frames per second, and the network input is a sequence of ten frames for which the model generates ten frame predictions. The layerwise weights λ_l were set to impose a smaller penalty for higher-level layers ($\lambda_0 = 1, \lambda_{>0} = .1$), which according to Lotter et al. (2016) results in better predictions. All timestep weights λ_t were set to zero, except for the first timestep.

3.3 Extracting spatiotemporal representations

We follow the same approach described by Lotter et al. (2016) to extract representations from each layer of the predictive coding model. For each sequence of ten frames in the input, the R_l activations are read for each layer l , which are then spatial pooled to match the higher-level layer dimensions and concatenated to form one tensor representation with dimensions (16, 20, 339) corresponding to a one-second spatiotemporal pattern. The idea of using "deep" representations that include activations from all layers is similar to recent unsupervised learning approaches in natural language processing such as Embeddings from Language Models (ELMo) (Peters et al., 2018). We assume that each layer might learn representations that reflect different timescales and the downstream classification model can learn to weight each layer according to the specific task. In Section 4.4, we report results for action classifiers using features from different layers.

3.4 Supervised action recognition

Once the one-second predictive coding representations were extracted, each moment representation corresponding to one second was flattened and used as input to an action classifier (Figure 3.3). As we are focused on the relative quality of representations and not state-of-the-art performance, we chose a simple linear support vector machine (SVM) (Cortes and Vapnik, 1995) to compare different predictive coding models. Hence, these classification experiments use linear separability as a criterion to evaluate representations. Since the linear SVM is a non-probabilistic model, video-level predictions are made by majority voting of predictions for each one-second feature.

Additionally, a recurrent neural network classifier was used to assess the importance of modelling longer timespans. In this case, a Long Short-Term Memory (LSTM) layer (Hochreiter and Schmidhuber, 1997) with 64 hidden units received a sequence of five overlapping one-second predictive coding representations totalling three seconds. The LSTM layer is followed by a fully connected layer with a softmax activate that outputs a probability distribution over the classes. Table 3.2 shows the architecture of the neural network classifier.

Layer name	Output shape
Input	(5,16,20,339)
Flatten	(5,108480)
LSTM	(64)
Dense	(number of classes)
Softmax	(number of classes)

Table 3.2: Recurrent neural network classifier layers.

3.5 Sensory substitution

The human brain possesses an extraordinary capacity to adapt its anatomical and functional structure in response to environmental changes, which is a phenomenon known as neuroplasticity in the neuroscience literature (Draganski et al., 2004; Maguire et al., 2000). One of the most exciting practical applications that leverage this property is the possibility of compensating sensory loss by encoding sensory data in a format that can be read by another peripheral sensory organ. For instance, recent work has shown the applicability of encoding visual information into sounds and tactile information to help blind people to see objects (Bach-y Rita and Kercel, 2003; Stiles and Shimojo, 2015).

Inspired by these sensory substitution experiments, we propose the use of the same predictive coding network designed for action recognition from visual stimuli to process audio data. The idea is to encode the audio information in a format that takes advantage of the capacity of predictive coding networks to capture spatiotemporal patterns, as demonstrated by experiments in sections 4.1 and 4.2. We build upon previous work that uses audio spectrograms images and convolutional neural networks for speech processing (Abdel-Hamid et al., 2014; Zhang et al., 2017), and impose a

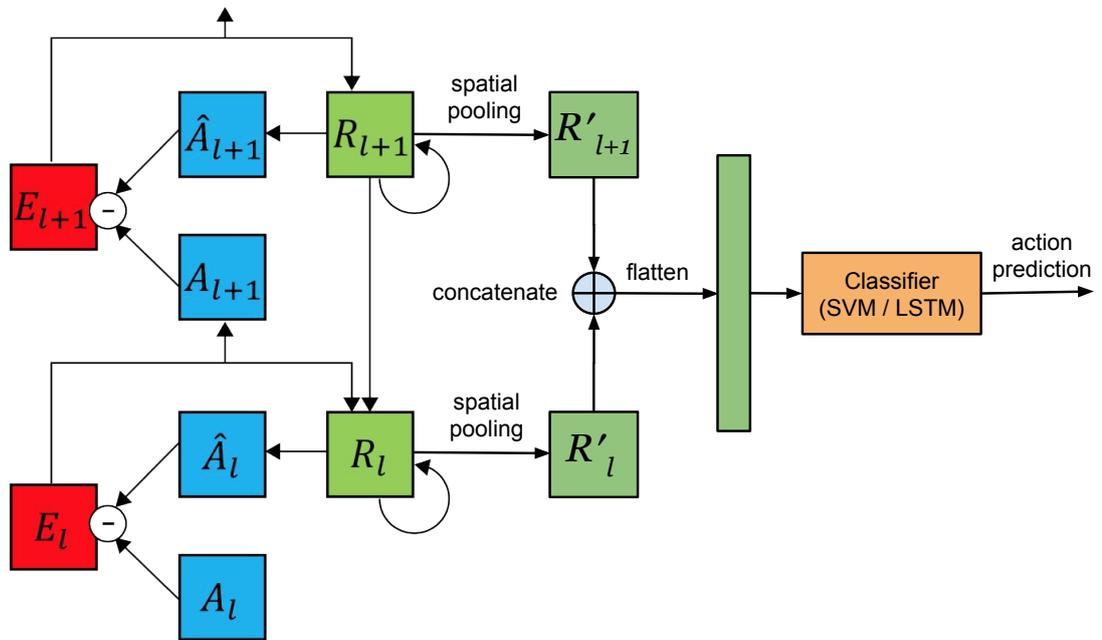


Figure 3.3: Action classification architecture showing two predictive coding layers. Representations R_l for each layer are concatenated after a spatial pooling operation. The resulting tensor is flattened and passed as input to an action classifier. Adapted from Lotter et al. (2016) Figure 1.

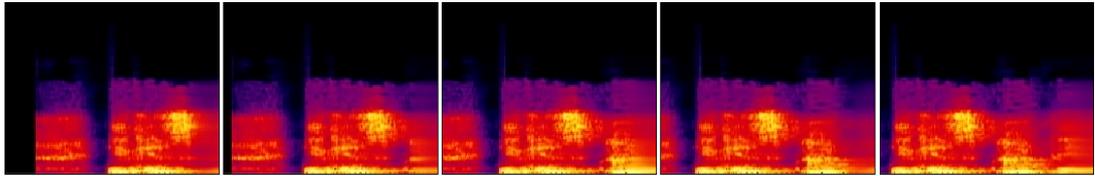


Figure 3.4: A sequence of spectrogram frames.

spatiotemporal coherence between subsequent audio frames. The resulting audio representation is a video showing a scrolling spectrogram with the same duration as the original video. Thus, we recast the audio classification problem as a spatiotemporal pattern recognition which is suitable for the predictive coding architecture.

To generate the scrolling spectrograms (Figure 3.4) we used `ffmpeg` tool (Bellard et al., 2000) `showspectrum` filter with Hanning window function (Harris, 1978) and overlap equal to zero. The generated videos had the same dimension and were sampled at the sample frame rate as the visual data so that we could use precisely the same neural network architecture, feature extraction and training procedure applied to the visual modality.

Chapter 4

Experiments

In this chapter, we detail the experiments used to assess the quality of predictive coding representations. We first pre-train the predictive coding model on varying sizes of unlabelled video datasets and evaluate the learned representations on small-scale action recognition tasks. Then, we investigate a novel application of the same predictive coding architecture to learn representations from audio spectrograms. Finally, we perform a transfer learning experiment and evaluate the audio and video representations on a widely used action recognition benchmark, the UCF-101 dataset (Soomro et al., 2012).

4.1 Next-frame predictions

According to Lotter et al. (2016), to predict the next frames a model needs to build an internal model that explains the movements of objects present in a given scene. Therefore, the most straightforward method to evaluate learned representations is to measure the quality of the generated predictions. Since the evaluation of generative models is a complex subject by itself (Theis et al., 2015), we follow Lotter et al. (2016) and use simply the mean squared error (MSE) between the predicted and actual frames as a quantitative measurement of prediction fidelity.

After training the predictive coding models as described in Section 3.2, we calculate the mean square error of predictions on a held-out dataset of 1000 videos spanning 10 action classes, as well as the results for a naive baseline that merely copies the last frame. The results in Table 4.1 show that as we add more data, the generative model yields better frame predictions, reducing the MSE error by 29.8% relative to the baseline. Also, it is clear that the improvement starts to plateau, giving only about 3% of

improvement when increasing the dataset size from three hours to 67 hours of video. However, it is worth to note that the Moments 67-hour dataset has videos from 200 different classes, including spatiotemporal information that differ significantly from the 10-class evaluation set. Therefore, we observe that the model continues to learn even when exposed to out-of-domain spatiotemporal data.

Model name	MSE	Relative change
Copy last frame	0.00795	0
PredNet random	0.14422	+1711.3%
PredNet KITTI	0.00816	+2.6%
PredNet Moments 3h	0.00581	-26.9%
PredNet Moments 67h	0.00558	-29.8%

Table 4.1: Average frame prediction errors (MSE) for different pre-trained models on a held-out set of 1000 videos spanning 10 action classes. Relative changes are computed in relation to the "copy last frame" baseline.

Inspecting sample frame predictions is also useful to qualitatively analyse how the model generalises spatiotemporal concepts. Figure 4.1 shows the final five frame predictions from a ten-frame sequence sampled from the *exercising* class. Interestingly, the predictions given by a predictive coding network with random weights (second row) are not random and seem to copy some of the most salient features of the previous frame without any colour information preserved. This observation illustrates the power of the predictive coding inductive bias and also explain why the randomly initialised model performs surprisingly well on action recognition tasks (see sections 4.2, 4.3, and 4.4) despite the poor performance regarding the MSE metric reported in Table 4.1.

The frames generated by the model trained on the KITTI dataset (third row) accurately predict the colour information but exhibit a significant amount of blurriness in the moving portions of the image (e.g., the woman’s legs). Most importantly, there is still a strong bias towards copying the contents of the previous frame. On the other hand, the predictions from the model trained on 67 hours of the Moments in Time dataset (fourth row) are less blurry and show a remarkable capacity of generalisation of scene dynamics. If we observe the position of the woman’s knees relative to the workout top in the last two frame predictions, we verify that the KITTI model prediction is very similar to the previous frame, while the Moments model extrapolates and

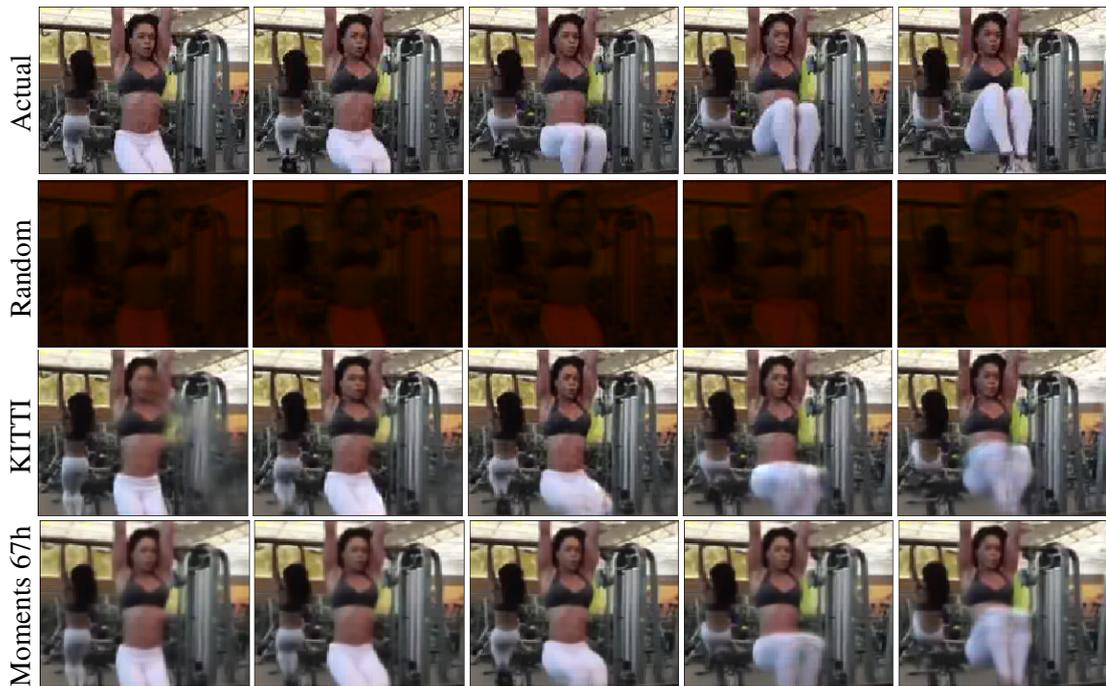


Figure 4.1: Last five frame predictions from a 10-frame timestep sequence sampled from the *exercising* class. First row shows ground truth frames. Second row shows predictions for a predictive coding model with random weights. The last two rows show predictions for models trained on KITTI dataset, and on 67 hours of videos from the Moments in Time dataset respectively. KITTI model predictions are blurrier and tend to be more similar to the previous frame.

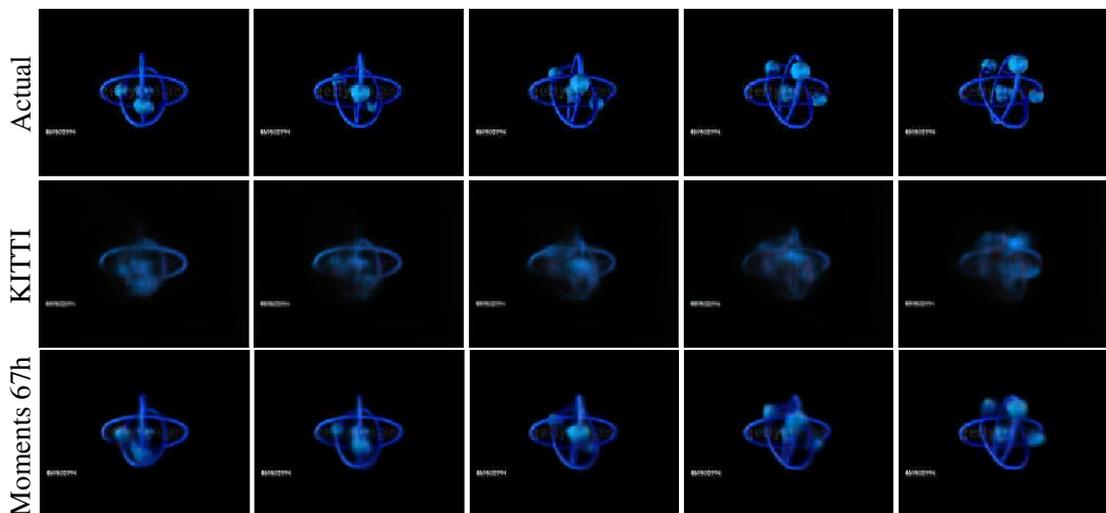


Figure 4.2: Last five frame predictions from a 10-frame timestep sequence sampled from the *twisting* class. First row shows ground truth frames. Second and third rows show predictions for predictive coding models trained on KITTI dataset and trained on 67 hours of videos from the Moments in Time dataset respectively.

”imagines” that legs should be at a higher position, much closer to the actual frame. For synthetic images (Figure 4.2), the difference in quality between the model trained on KITTI and Moments 67 hours is more pronounced, with significant blurriness in the first case.

4.2 Small-scale action recognition

While the next-frame prediction analysis from Section 4.1 indicates that the predictive coding model can generalise spatiotemporal patterns, there is still no evidence that the learned representations are transferable to other domains. To address this question, we investigate the usefulness of predictive coding representations to decode high-level action concepts such as *walking*, *exercising*, and *cooking*. In particular, we are interested in scenarios in which there is a limited amount of labelled data that are expensive to obtain, and a large number of unlabelled videos cheaply available for the unsupervised predictive coding model. In these cases, the unsupervised pre-training can be viewed as a regulariser for the supervised task. (Goodfellow et al., 2016).

Intuitively, we know that some of the actions are readily determined by the objects that appear in the scene while other events require fine-grained distinction of object dynamics. For instance, if objects such as food and cookware are identified in a given frame, one could readily infer that the activity is *cooking*. In contrast, to differentiate between *walking* and *running*, the details of the temporal dynamics of the entities present in the scene are crucial, making the classification task very challenging for current video understanding models. In fact, most of the state-of-the-art action classification approaches rely on hand-crafted features such as optical flow to encode temporal information (Carreira and Zisserman, 2017).

Our small-scale experiment uses 100, 50, and 100 videos per class for training, validation, and testing respectively. We create two binary classification tasks, which are intentionally designed to explore different challenges of action perception discussed above. The binary *spatial* task consists in classifying videos from *cooking* and *walking* classes, which should be effortlessly distinguishable just by identifying the entities involved in the action. Similarly, the binary *temporal* task uses the target actions *running* and *walking*, which should require a subtler perception of scene dynamics. We also report results for classification tasks involving ten classes. The unsupervised predictive coding model is allowed to use the training split but never touches the validation and test sets.

In the linear SVM classifier, we use the squared hinge loss, a penalty parameter $C = 1.0$, stopping tolerance $1e-4$, and L2 regularisation. The neural network classification models used an LSTM layer with 512 hidden units followed by a fully connected layer. cross-entropy loss for training and the parameters were optimised using the Adam gradient-based algorithm (Kingma and Ba, 2014). Since we are dealing with small datasets and high-dimensional moment representations, an aggressive dropout regularisation of 0.9 was applied (Srivastava et al., 2014). Early stopping was used to stop training after the validation loss stopped improving for ten epochs. The SVM and LSTM classifiers were implemented using the scikit-learn (Pedregosa et al., 2011) and Keras (Chollet et al., 2015) open-source libraries respectively.

Features + Classifier	2-class spatial	2-class temporal	10-class
VGG random + SVM	67.0	56.0	18.7
VGG ImageNet + SVM	85.5	67.0	52.8
VGG ImageNet + LSTM	87.4	58.4	43.2
PredNet random + SVM	67.6	62.6	30.1
PredNet KITTI + SVM	73.2	70.7	39.8
PredNet Moments 3h + SVM	73.2	66.1	39.5
PredNet Moments 67h + SVM	74.2	65.1	41.4
PredNet Moments 67h + LSTM	81.6	55.8	42.9

Table 4.2: Classification accuracies (percentage) for different pre-trained models on a held-out set of 100 videos per class. First set of rows list results for a baseline model using features extracted from the VGG16 convolutional image classifier (Simonyan and Zisserman, 2014b).

Classification accuracies are listed in Table 4.2, which also include baseline models using features extracted from a VGG16 model (Simonyan and Zisserman, 2014b) pre-trained on the ImageNet dataset (Krizhevsky et al., 2012). In the following paragraphs, we discuss the most relevant results.

When models pre-trained on Imagenet fail Confirming the findings of previous work (Carreira and Zisserman, 2017), pre-training models on ImageNet gives a substantial improvement in classification performance, especially on the binary spatial and 10-class tasks, for which the identification of objects is determinant. However, there is

a significant drop in accuracy on the binary temporal task, which indicates the model falls short of capturing fine-grained temporal patterns needed to distinguish between *running* and *walking* actions. For this reason, many of the state-of-the-art approaches for action recognition are based on "two-stream" models (Simonyan and Zisserman, 2014a; Carreira and Zisserman, 2017), which integrate hand-crafted temporal features such as optical flow and raw RGB frames.

Predictive coding temporal inductive bias Representations generated by predictive coding models with random weights outperform random VGG features in all tasks, which indicates that predictive coding incorporates inductive biases that are better suited to spatiotemporal perception. Remarkably, the features extracted from the predictive coding model trained on only 41K unlabelled frames from the KITTI dataset gives the best performance on the binary temporal task, outperforming by a significant margin the VGG model pre-trained on more than one million labelled images (Simonyan and Zisserman, 2014b).

The more data, the better As we add more data to the predictive coding model, the performance on the binary spatial task improves while the accuracy on the binary temporal task falls consistently. We believe this performance drop is due to the introduction of a large number of unrelated classes in the 3-hour (10 classes) and 67-hour (200 classes) versions, as the performance on the 10-class seems to improve nicely with larger datasets. Nevertheless, a more careful experiment would be required to confirm this hypothesis.

Modelling longer time spans In our experiments, the LSTM models use sequences of five VGG/moment representations spanning three seconds of spatiotemporal data, which results in better performance on the binary spatial task compared to a simple average of SVM predictions for each representation. On the other hand, performance is severely affected on the binary temporal task for both types of representations. Again, further investigation is needed to understand the reason behind these results.

4.3 Exploring the audio modality

After applying the sensory substitution trick proposed in Section 3.5, we obtained two datasets with 2 hours and 37 hours of auditory data. We then performed the same

unsupervised training procedure used for the video modality (Section 3.2) and found the performance pattern in the frame-prediction evaluation was similar to the visual modality results (see Table 4.3). The relative reduction of the best predictive coding model with respect to the naive baseline was 92.4% (versus 29.8 for the video modality), which suggests that the prediction of audio spectrograms is easier than predicting frames in natural images. This idea is also supported by samples of predicted frames (Figure 4.3), which clearly show a pronounced difference in quality between prediction from the models trained on the KITTI and Moments datasets. We believe the low resolution of the spectrogram frames and the simple dynamics of the spatiotemporal pattern (merely scrolling from right to left) are decisive factors for the excellent quality of predictive coding predictions.

Regarding the small-scale action recognition experiment (Section 4.2), the results in Table 4.4 show that adding more training data to the unsupervised training step leads to improvement on both binary tasks. On the 10-class task, there was no improvement in accuracy over the model with random weights, which gave a surprisingly strong baseline score. Notably, all predictive coding models followed by an SVM classifier performed better than the classifiers based on features from a VGG model pre-trained on ImageNet. In this case, the weights learned from an image recognition task do not generalise well to the spectrograms domain, and the inductive biases provided by the predictive model proved to be useful. Due to the small amount of data and extreme overfitting, LSTM classifiers were not used in this experiment.

Model name	MSE	Relative change
Copy last frame	0.011193	0
PredNet random	0.079011	+605.9%
PredNet KITTI	0.011392	+1.8%
PredNet Moments 2h	0.000930	-91.7%
PredNet Moments 37h	0.000856	-92.4%

Table 4.3: Average frame prediction errors (MSE) for different pre-trained models on a held-out set of 622 videos spanning 10 action classes. Relative changes are computed in relation to the "copy last frame" baseline.

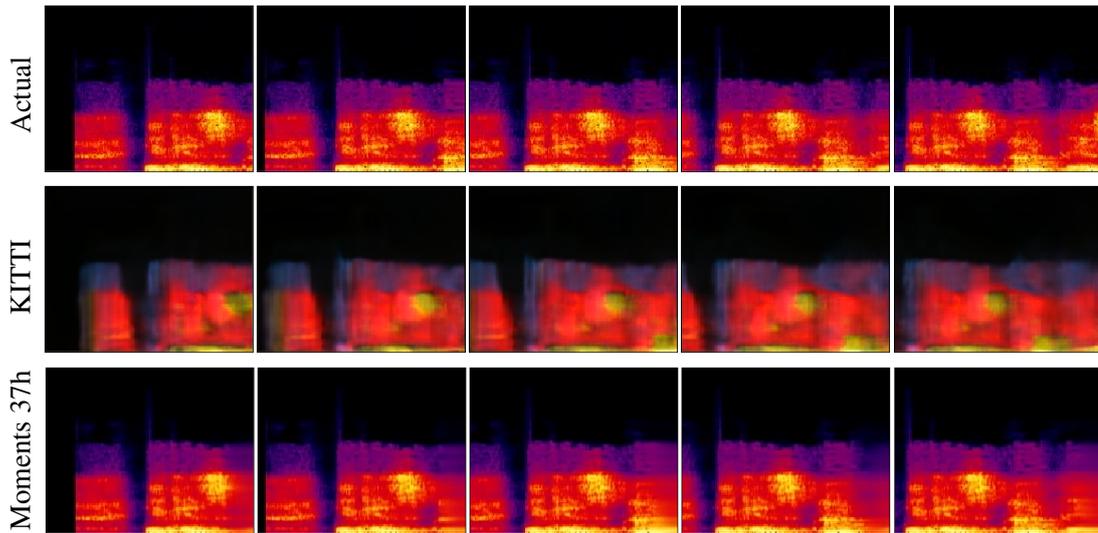


Figure 4.3: Last five spectrogram frame predictions from a 10-frame timestep sequence sampled from the *speaking* class. First row shows ground truth frames. Second and third rows show predictions for predictive coding model trained on KITTI dataset and trained on 37 hours of videos from the Moments in Time dataset respectively. Predictions for the model with random weights are omitted as they are mostly black frames.

4.4 Transfer learning

All the experiments so far were based on a small subset of the Moments in Time dataset specifically designed to test our hypothesis concerning the effects of different amounts of pre-training data on various tasks. One question that remains to be explored is how predictive coding representations compare to other approaches in action recognition literature. To address this question, we chose the UCF-101 dataset (Soomro et al., 2012), a widely used action recognition benchmark with 13,320 video clips spanning 101 action classes. Besides the vast amount of published results, this dataset is suitable for our experiment because it contains only 27 hours of video, which is still manageable for training without expensive distributed computing across several GPUs.

In this experiment, we used our best predictive coding models pre-trained on 67 hours of visual and 37 hours of auditory information from Moments in Time videos with no further fine-tuning. Thus, the task requires the model to generalise patterns learned from the Moments in Time dataset to a new dataset with unseen spatiotemporal concepts.

Features + Classifier	2-class spatial	2-class temporal	10-class
VGG ImageNet + SVM	57.9	53.0	24.7
PredNet Audio random + SVM	63.6	56.8	30.3
PredNet Audio KITTI + SVM	62.0	50.8	29.4
PredNet Audio Moments 2h + SVM	66.9	56.8	29.1
PredNet Audio Moments 37h + SVM	67.8	58.3	30.0

Table 4.4: Classification accuracies (percentage) for different pre-trained models on a held-out set of 60 videos per class. First set of rows list results for a baseline model using features extracted from the VGG16 convolutional image classifier (Simonyan and Zisserman, 2014b).

4.4.1 Next-frame predictions

Again, we resort to next-frame predictions to get a qualitative assessment of the model understanding of action dynamics, namely for the *CliffDiving* action from UCF-101. As shown in Figure 4.4, while the model can predict the overall changes such as camera movement that causes the occlusion of the platform, guessing the detailed pattern of the diver’s body is difficult, and the prediction degenerates to a blurry blob. Besides the multiple hidden causes of movement, this example introduces extra challenges such as the complex and swift movements in small portions of the image (the diver’s acrobatics). To isolate these factors, we chose another sample that shows only the body movements occupying a larger portion of the image and no camera shifts. The predictions in Figure 4.5 show that the model still falls short of predicting the body movements, suggesting that rapid image changes are particularly difficult for the predictive coding model. Future work to address this issue might include more training data containing complex body movements and working with frame rates higher than we used in these experiments (10 frames per second).

4.4.2 Action recognition

Spatiotemporal representations were extracted as described in Section 3.3 and a recurrent neural network with one LSTM layer (64 hidden units) followed by a fully connected layer was used as an action classifier. The classifier received as input sequences of five predictive coding representations corresponding to 3 seconds of video.

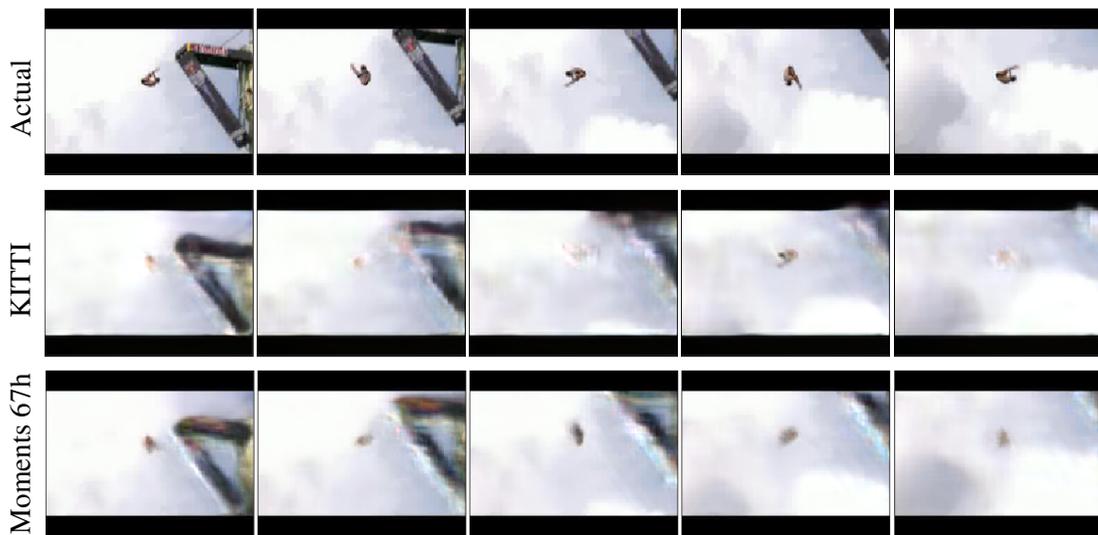


Figure 4.4: Last five spectrogram frame predictions from a 10-frame timestep sequence sampled from the *CliffDiving* class. First row shows ground truth frames. Second and third rows show predictions for predictive coding model trained on KITTI dataset and trained on 67 hours of videos from the Moments in Time dataset respectively. The model captures the overall camera movement and the position of the diver but falls short of figuring out the finer details.

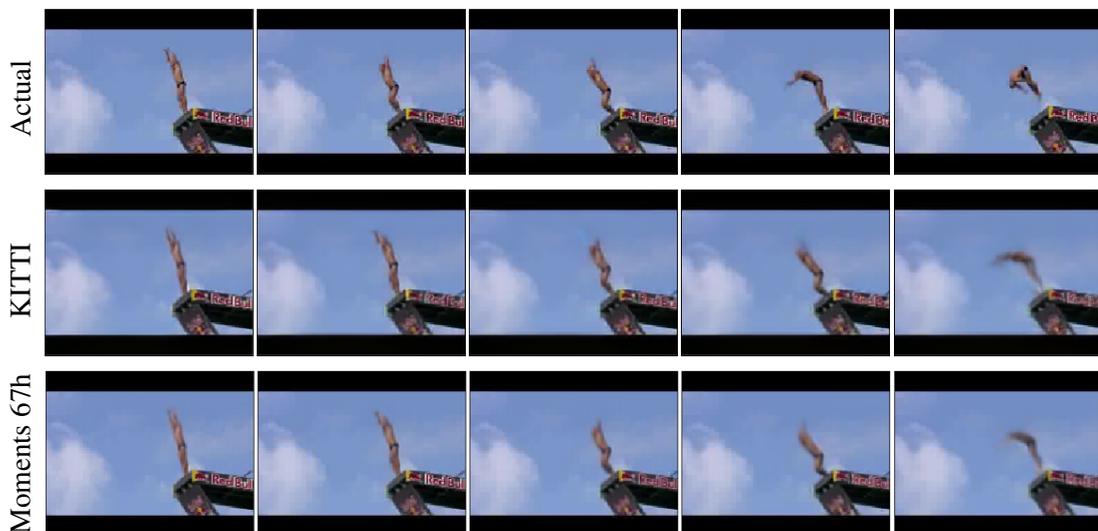


Figure 4.5: Last five spectrogram frame predictions from a 10-frame timestep sequence sampled from the *CliffDiving* class. First row shows ground truth frames. Second and third rows show predictions for predictive coding model trained on KITTI dataset and trained on 67 hours of videos from the Moments in Time dataset respectively. Even without camera movements, the model fails to predict the body movements and tends to copy the previous frames.

For videos with more than 3 seconds of duration, the final video-level predictions were the average of all 3-second predictions (for more details refer to Section 3.4). Table 4.5 lists the top-1 accuracies on the test set of UCF-101 first split, which contains 9537 clips for training and 3783 clips for testing.

Features + Classifier	Accuracy (%)	Pre-training dataset
PredNet Video random + LSTM	1.64	-
PredNet Video 67h + LSTM	51.9	Moments in Time
CNN tuple verification	50.2	UCF-101
Inception + LSTM	54.2	-

Table 4.5: Accuracies (top-1 percentage) for different pre-trained models on test set of UCF-101 split 1. We also include results for the CNN tuple verification (Misra et al., 2016) and an LSTM classifier trained on top an Inception convolutional network trained from scratch (Carreira and Zisserman, 2017).

As opposed to the small-scale classification experiments in Sections 4.2 and 4.3, the predictive coding model with random weights gives a poor performance of 1.64%, which is slightly above the random baseline. However, when we train the classifier with features generated by the 67-hour predictive coding model, the accuracy increases to 51.9%, which is competitive with results from the unsupervised "tuple verification" by Misra et al. (2016) and an LSTM classifier using the Inception convolutional network (Carreira and Zisserman, 2017). It is worth to note however that in both of these approaches, the convolutional models are fine-tuned end-to-end using the UCF-101 labels. In our case, the predictive coding weights were kept fixed and only the weights from the LSTM classifier were optimized for the specific task.

We also trained the auditory predictive coding model on the 51 action classes of the UCF-101 dataset that contains audio information. The top-1 accuracy results are reported in Table 4.6. As expected, the audio information is much less useful to distinguish action classes, as many videos have soundtracks and other kinds of audio data that are completely unrelated to the activity. Still, there was a significant improvement from the classifier trained on the features generated by the random-weights model to the classifier based on the 37-hour pre-trained model. For comparison, we also report the results of the CaffeNet version by Wang et al. (2016), which is a convolutional network trained on spectrograms that have no spatial coherence across frames.

Remarkably, our simple one-layer LSTM classifier is competitive with their complex convolutional model trained end-to-end using action class labels, which demonstrates the generality of predictive coding representations.

The simple average predictions of visual and auditory models (last row of table 4.7) gives a slight improvement over the video-only model, indicating that the audio representations may capture complementary information that is useful for the task. However, further work is required to demonstrate that this performance difference is statistically significant.

Features + Classifier	Accuracy (%)	Pre-training dataset
PredNet Audio random + LSTM	22.7	-
PredNet Audio 37h + LSTM	24.8	Moments in Time
Caffenet (Wang et al., 2016)	25.2	-

Table 4.6: Accuracies (top-1 percentage) for different pre-trained models on test set of UCF-101 split 1 (only videos from the 51 classes that contain audio).

A critical property of good representations introduced in Section 2.1 is that they become more abstract and invariant in the upper layers. To verify if predictive coding representations exhibit this characteristic, we trained the same LSTM classifier using features extracted from different layers of the pre-trained predictive coding model. Indeed, the results listed in Table 4.7 show an increase in performance for higher-level layers. Moreover, all features individually produce worse results than the merged representations of all layers, which suggests the classifier captured complementary information from the different levels of abstraction. These results are also in consonance with recent unsupervised learning experiments in natural language processing (Peters et al., 2018).

Features + Classifier	Accuracy (%)
PredNet Video all layers + LSTM	51.9
PredNet Video layer 0 + LSTM	12.8
PredNet Video layer 1 + LSTM	31.1
PredNet Video layer 2 + LSTM	34.3
PredNet Video layer 3 + LSTM	40.4
PredNet Video + Audio	52.4

Table 4.7: Accuracies (top-1 percentage) for models trained using representations from different layers of a PredNet pre-trained on 67 hours of visual data. PredNet Video + Audio is an ensemble with the predictive coding model pre-trained on 37 hours of auditory data. Results are reported for the test set of UCF-101 split 1.

Chapter 5

Discussion and Future work

Our empirical analysis supports the hypothesis of predictive coding as a powerful inductive bias for learning common sense by observing how the world dynamics unfold. The simple predictive coding architecture proposed by Lotter et al. (2016) can predict future frames in a way that suggests a good level of generalisation across different situations, surpassing by a significant margin a baseline predictor that copies the previous frame. Additionally, next-frame analysis proved to be a useful tool to inform the investigation of improvements to the model. For instance, we found predictions degraded when fast-moving entities appear in the scene, suggesting that temporal resolution used in training was not adequate.

From the action recognition experiments, we learned that for some activities such as *running* and *walking*, the perception of fine-grained dynamics is fundamental, and in these cases, the predictive coding model has an edge on the traditional convolutional approaches. However, there are many cases in which the identification of the entities present is informative enough to determine the action. For instance, the *cliff diving* activity shown in Figures 4.4 and 4.5 could be readily identified as diving by a human even by inspecting a single frame. For this reason, the state-of-the-art action recognition approaches will continue to rely on pre-trained image classifiers. However, there is an opportunity to replace the handcrafted temporal streams (Carreira and Zisserman, 2017) with data-driven approaches such as the predictive coding model.

In the out-of-domain action classification using the UCF-101 dataset (Wang et al., 2016), a simple one-layer LSTM classifier on top of predictive coding representations performed on par with deep convolutional models fine-tuned to labelled video data (Misra et al., 2016; Carreira and Zisserman, 2017). Crucially, the sensory substitution trick we proposed also performed on par with convolution classifiers fine-tuned to the

auditory information of the UCF-101 dataset. This result opens an exciting alternative towards homogeneous multimodal architectures endowed with neuroplastic properties (Draganski et al., 2004).

While our experimental results are far from state-of-the-art action recognition models, our approach learns representations that are potentially more general and transferable to other tasks for which annotated data is expensive to obtain, since we do not fine-tune the unsupervised model using task-specific labels. Nevertheless, the experimental data revealed limits in the predictive coding model's capacity to learn fine-grained spatiotemporal patterns. In the next sections, we discuss further research direction that may help to address these issues.

5.1 Scaling predictive coding training

One of the most important conclusions of our experiments is that the quality of predictive coding representations improve as we add more data in the pre-training step. In fact, our larger pre-training dataset of 67 hours of video (2.4 million frames) despite being around 60 times larger than the original implementation by Lotter et al. (2016), it is small compared to many recent action recognition datasets such as the full Moments in Time dataset with one million videos (over 800 hours) (Monfort et al., 2018) or the Sports-1M dataset (Karpathy et al., 2014).

Even for lower dimensional natural language processing data, successful unsupervised learning approaches consume a vast amount of computing resources, with pre-training steps lasting over one month on eight GPUs (Radford et al., 2018). Unfortunately, our current Keras-based implementation is not optimised for parallelisation and to further improve the performance given by predictive coding pre-training, a full re-engineering is required to leverage more recent Tensorflow (Abadi et al., 2015) APIs that deliver more efficient parallelisation and input pipelines.

During our experiments, it became clear that engineering scalable pipelines can be a severe bottleneck for scientific research in a highly empirical field such as machine learning. To illustrate, a recent publication by Jia et al. (2018) reported a flummoxing ImageNet (Deng et al., 2009) training procedure across 2048 GPUs achieving a top-1 test accuracy of 75.8% in only 6.6 minutes. While these results are impressive, they also invite to reflect on the increasing gap between industry and academia and its potential negative consequences in future research.

5.2 Multimodal predictive coding

In this work, the visual and auditory models were trained separately and their predictions combined straightforwardly. However, experimental results in neuroscience suggest that the interaction between modalities is much more complex and can occur even at early stages of visual and auditory cortical processing (Ghazanfar and Schroeder, 2006; Falchier et al., 2002). The predictive coding architecture can be extended to explore such forms of early fusion (Snoek et al., 2005). Since the layer predictions already integrate representations from errors from the current layer and representation from upper layers, would be straightforward to add representations from other modalities at each level. This extension would allow empirical priors (Friston, 2010) learned from each modality condition next-frame predictions, resulting in potentially more efficient multisensory integration.

As already suggested in our project proposal, we hypothesise that multimodal tasks involving videos such as cross-modal retrieval (Aytar et al., 2017) and grounded language learning (Hermann et al., 2017) can benefit from a more sophisticated sensory integration model. In future work, we intend to explore the application of predictive coding representations in such problems.

5.3 Integrating action

One of the most important consequences of the free-energy principle is that action is naturally integrated with perception and learning. To minimise the variational free-energy, the agent not only has to improve its perceptual capabilities by also act to improve its model of the world (see Equation 2.20) (Friston and Kiebel, 2009; Clark, 2013). Furthermore, under this framework, the agent has to sample the data, or more generally, act to realise predictions "that are biased toward preferred outcomes" (Friston et al., 2017), a concept known as active inference. If the multimodal experiments suggested in Section 5.2 are successful, the next step would be to investigate the feasibility of modelling actions and even attentional mechanisms (Koelewijn et al., 2010) as additional modalities.

5.4 Learning intuitive physics

In our supervised action recognition experiments, we used datasets that contain high-level human-centred classes such as *speaking* and *dancing*. Although the experimental results showed a reasonable level of generalisation across different actions, exposing the network to lower-level physical concepts may result in better representations. For instance, the Something-Something dataset (Goyal et al., 2017) is a collection of videos centred on everyday actions labelled with structured natural language labels such as "Putting [*something*] into [*something*]" or "Opening [*something*]". Thus, a model trained in this dataset is encouraged to reason about physical primitives including spatial constraints and object affordances, which could offer a better level of abstraction for common sense learning.

Chapter 6

Conclusion

In this dissertation, we investigated a neuro-inspired model for spatiotemporal representation learning, known as predictive coding (Friston and Kiebel, 2009; Rao and Ballard, 1999). Our goal was to develop models that can learn common sense from a large number of unlabelled videos and transfer this knowledge to other video understanding tasks. In particular, we extend previous work by Lotter et al. (2016) to evaluate predictive coding representations on action recognition tasks.

We reviewed the theory behind machine learning models capable of learning economic and transferable representations that disentangle hidden causal factors. We showed that minimising the description length of data corresponds to minimising free energy or, under the free-energy principle (Friston, 2010), minimise the surprise on sensory states. This information-theoretic "coincidence" establishes a fascinating relationship among machine learning models, statistical thermodynamics, and living organisms that resist a natural tendency to disorder.

Moreover, we trained and evaluated the unsupervised predictive coding model by Lotter et al. (2016) in various experimental settings. Our results suggest that predictive coding representations improve with increasing amount of unlabelled data and that they are useful to discriminate between higher-level action concepts. Notable, we found that the same predictive coding architecture can learn from auditory data and generalise across different datasets.

Finally, we propose further improvements regarding scaling predictive coding models to larger datasets and implementing more sophisticated multimodal learning by using cross-modal empirical priors. We believe that this neuro-inspired representation learning algorithm is a promising approach towards the implementation of machines that learn common sense while maximising the evidence of their own existence.

Bibliography

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Abdel-Hamid, O., Mohamed, A.-r., Jiang, H., Deng, L., Penn, G., and Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545.
- Aytar, Y., Vondrick, C., and Torralba, A. (2017). See, hear, and read: Deep aligned representations. *arXiv preprint arXiv:1706.00932*.
- Bach-y Rita, P. and Kercel, S. W. (2003). Sensory substitution and the human–machine interface. *Trends in cognitive sciences*, 7(12):541–546.
- Baillargeon, R., Spelke, E. S., and Wasserman, S. (1985). Object permanence in five-month-old infants. *Cognition*, 20(3):191–208.
- Barber, D. (2012). *Bayesian reasoning and machine learning*. Cambridge University Press.
- Bellard, F. et al. (2000). Ffmpeg. <https://www.ffmpeg.org>. Accessed Aug 01 2018.
- Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

- Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155.
- Bengio, Y., Lamblin, P., Popovici, D., and Larochelle, H. (2007). Greedy layer-wise training of deep networks. In *Advances in neural information processing systems*, pages 153–160.
- Buckley, C. L., Kim, C. S., McGregor, S., and Seth, A. K. (2017). The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*.
- Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE.
- Chollet, F. et al. (2015). Keras. <https://keras.io>. Accessed Aug 01 2018.
- Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, 36(3):181–204.
- Clark, A. (2017). How to knit your own markov blanket.
- Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.
- Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. (1995). The helmholtz machine. *Neural computation*, 7(5):889–904.
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee.
- Draganski, B., Gaser, C., Busch, V., Schuierer, G., Bogdahn, U., and May, A. (2004). Neuroplasticity: changes in grey matter induced by training. *Nature*, 427(6972):311.
- Falchier, A., Clavagnier, S., Barone, P., and Kennedy, H. (2002). Anatomical evidence of multimodal integration in primate striate cortex. *Journal of Neuroscience*, 22(13):5749–5759.
- Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2):127.

- Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., and Pezzulo, G. (2017). Active inference: a process theory. *Neural Computation*, 29(1):1–49.
- Friston, K. and Kiebel, S. (2009). Predictive coding under the free-energy principle. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 364(1521):1211–1221.
- Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. (2013). Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237.
- Ghazanfar, A. A. and Schroeder, C. E. (2006). Is neocortex essentially multisensory? *Trends in cognitive sciences*, 10(6):278–285.
- Glorot, X., Bordes, A., and Bengio, Y. (2011). Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 513–520.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>.
- Goyal, R., Kahou, S. E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Freund, I., Yianilos, P., Mueller-Freitag, M., et al. (2017). The something something video database for learning and evaluating visual common sense. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 1, page 3.
- Harris, F. J. (1978). On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66(1):51–83.
- Hermann, K. M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari, D., Czarnecki, W. M., Jaderberg, M., Teplyashin, D., et al. (2017). Grounded language learning in a simulated 3d world. *arXiv preprint arXiv:1706.06551*.
- Hinton, G. E., Dayan, P., Frey, B. J., and Neal, R. M. (1995). The “wake-sleep” algorithm for unsupervised neural networks. *Science*, 268(5214):1158–1161.
- Hinton, G. E. and Zemel, R. S. (1994). Autoencoders, minimum description length and helmholtz free energy. In *Advances in neural information processing systems*, pages 3–10.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

- Jia, X., Song, S., He, W., Wang, Y., Rong, H., Zhou, F., Xie, L., Guo, Z., Yang, Y., Yu, L., et al. (2018). Highly scalable deep learning training system with mixed-precision: Training imagenet in four minutes. *arXiv preprint arXiv:1807.11205*.
- Jurafsky, D. and Martin, J. H. (2014). *Speech and language processing*, volume 3. Pearson London.
- Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Koelewijn, T., Bronkhorst, A., and Theeuwes, J. (2010). Attention and the multiple stages of multisensory integration: A review of audiovisual studies. *Acta psychologica*, 134(3):372–384.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.
- LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.
- LeCun, Y., Touresky, D., Hinton, G., and Sejnowski, T. (1988). A theoretical framework for back-propagation. In *Proceedings of the 1988 connectionist models summer school*, volume 1, pages 21–28. CMU, Pittsburgh, Pa: Morgan Kaufmann.
- Lotter, W., Kreiman, G., and Cox, D. (2016). Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*.
- Maguire, E. A., Gadian, D. G., Johnsrude, I. S., Good, C. D., Ashburner, J., Frackowiak, R. S., and Frith, C. D. (2000). Navigation-related structural change in the hippocampi of taxi drivers. *Proceedings of the National Academy of Sciences*, 97(8):4398–4403.
- McMurray, B. (2007). Defusing the childhood vocabulary explosion. *Science*, 317(5838):631–631.

- McNaught, A. D. and McNaught, A. D. (1997). *Compendium of chemical terminology*, volume 1669. Blackwell Science Oxford.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Misra, I., Zitnick, C. L., and Hebert, M. (2016). Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer.
- Monfort, M., Zhou, B., Bargal, S. A., Andonian, A., Yan, T., Ramakrishnan, K., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., et al. (2018). Moments in time dataset: one million videos for event understanding. *arXiv preprint arXiv:1801.03150*.
- Narayanan, H. and Mitter, S. (2010). Sample complexity of testing the manifold hypothesis. In *Advances in Neural Information Processing Systems*, pages 1786–1794.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2018). Improving language understanding by generative pre-training.
- Rao, R. P. and Ballard, D. H. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79.
- Sharp, K. and Matschinsky, F. (2015). Translation of ludwig boltzmanns paper on the relationship between the second fundamental theorem of the mechanical theory of heat and probability calculations regarding the conditions for thermal equilibrium sitzungberichte der kaiserlichen akademie der wissenschaften. mathematisch-naturwissen classe. abt. ii, lxxvi 1877, pp 373-435 (wien. ber. 1877, 76: 373-435).

- reprinted in *wiss. abhandlungen*, vol. ii, reprint 42, p. 164–223, barth, leipzig, 1909. *Entropy*, 17(4):1971–2009.
- Simonyan, K. and Zisserman, A. (2014a). Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576.
- Simonyan, K. and Zisserman, A. (2014b). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Snoek, C. G., Worring, M., and Smeulders, A. W. (2005). Early versus late fusion in semantic video analysis. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 399–402. ACM.
- Soomro, K., Zamir, A. R., and Shah, M. (2012). Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.
- Spelke, E. S., Katz, G., Purcell, S. E., Ehrlich, S. M., and Breinlinger, K. (1994). Early knowledge of object motion: Continuity and inertia. *Cognition*, 51(2):131–176.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.
- Srivastava, N., Mansimov, E., and Salakhutdinov, R. (2015). Unsupervised learning of video representations using lstms. In *International conference on machine learning*, pages 843–852.
- Stiles, N. R. and Shimojo, S. (2015). Auditory sensory substitution is intuitive and automatic with texture stimuli. *Scientific reports*, 5:15628.
- Theis, L., Oord, A. v. d., and Bethge, M. (2015). A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.
- Wang, C., Yang, H., and Meinel, C. (2016). Exploring multimodal video representation for action recognition. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 1924–1931. IEEE.
- Wiatowski, T. and Bölcskei, H. (2018). A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, 64(3):1845–1866.

- Xingjian, S., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-c. (2015). Convolutional lstm network: A machine learning approach for precipitation now-casting. In *Advances in neural information processing systems*, pages 802–810.
- Zeiler, M. D. and Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer.
- Zhang, Y., Pezeshki, M., Brakel, P., Zhang, S., Bengio, C. L. Y., and Courville, A. (2017). Towards end-to-end speech recognition with deep convolutional neural networks. *arXiv preprint arXiv:1701.02720*.
- Zwanzig, R. (1973). Nonlinear generalized langevin equations. *Journal of Statistical Physics*, 9(3):215–220.