Prioritizing Compression Explains Human Perceptual Preferences

Francisco M. López
Frankfurt Institute for Advanced Studies and
Xidian-FIAS International Joint Research Center
lopez@fias.uni-frankfurt.de

Bertram E. Shi
Hong Kong University of Science and Technology
eebert@ust.hk

Jochen Triesch
Frankfurt Institute for Advanced Studies
triesch@fias.uni-frankfurt.de

Abstract

We present prioritized representation learning (PRL), a method to enhance unsupervised representation learning by drawing inspiration from intrinsic motivations and active exploration. PRL re-weights training samples based on an intrinsic priority function embodying preferences for certain inputs. We show how common human perceptual biases across different sensory modalities emerge through a priority function promoting compression. We also demonstrate that the same priority function establishes an early exposure effect where stimuli that are seen early in training become preferred. Our results reveal that PRL can mimic the results of active representation learning even in the absence of active control over the input.

1 Introduction

Throughout their lives, humans develop a range of perceptual preferences. While some of these are rather universal and shared with other animal species (e.g. finding symmetries aesthetically pleasing [1]), others are shaped by an individual's culture and/or personal experiences (e.g. a baby's joy when looking at their mom but not at strangers [2] or an adult's preference for foods they used to eat in their youth [3]). Are there general mechanisms driving the development of such preferences?

Humans and non-human animals interact with their environments and thus control the statistics of their sensory inputs. This gives them the opportunity to preferentially sample specific experiences at the expense of others. An initial preference for certain stimuli (even if small) may therefore result in preferential sampling of similar stimuli, which, under certain conditions, can further amplify this preference, establishing a positive feedback loop. From an evolutionary perspective, such a mechanism of preference development may prevent unnecessary risks associated with overly curious sampling of ever new experiences, which might waste precious resources on learning too many things (and none of them well). Such a mechanism of preference formation is also consistent with results on *perceptual fluency*, i.e. the idea that perceptual inputs that have been well-practiced and are easy to process are associated with positive affect [4, 1]. Crucially, while an active learner can selectively sample inputs based on its preferences, the opposite is true for passive learners, where such preferences may be hard to establish.

In this work we present prioritized representation learning (PRL), a model to explain the origins of common human perceptual preferences. Drawing inspiration from prioritized experience replay in reinforcement learning [5], we propose that standard (passive) representation learning can be enhanced by means of a priority function that re-weights inputs in a training batch according to

Accepted at the Intrinsically Motivated Open-ended Learning workshop at NeurIPS 2024.

Algorithm 1: Prioritized representation learning (PRL)

Input dataset $\mathbf{X} = \{\mathbf{x}_i\}$, priority weighting function $\mathcal{W}$, loss function $\mathcal{L}$, total epochs E, batch size N, learning rate γ
Initialize representation model parameters θ
for *epoch = 1,2, ...,E* **do**
  Sample batch $\mathbf{B} = \{\mathbf{b}_i\}$ from dataset: $\mathbf{B} \sim \mathbf{X}$
  for $\mathbf{b}_i = \mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_N$ **do**
    Compute element loss: $L_i = \mathcal{L}(\mathbf{b_i}|\theta)$
    Compute weighting: $w_i = \mathcal{W}(\mathbf{b_i}|\bar{\theta})$
  end
  Compute weighted loss: $\mathcal{L}^* = \sum_i w_i L_i \;/\; \sum_j w_j$
  Update representation model: $\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}^*$
end
Output trained representation model parameters θ

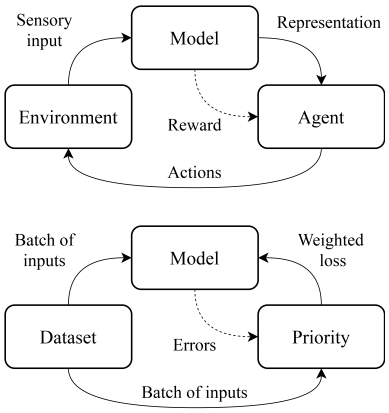


Figure 1: AEC sensorimotor loop (top) and PRL equivalent (bottom)

a generic objective function. This re-weighting can also be interpreted as filtering out irrelevant inputs in order to prioritize the representations of more relevant inputs, establishing a link to attention mechanisms. The method is agnostic with respect to the chosen representation model, as long as its loss function can be appropriately weighted. Thus, anything from a simple dimensionality reduction algorithm [6] to a contrastive learning encoder [7] may benefit from PRL. By way of example, we show how a deep autoencoder trained with an efficient coding priority favoring compressible inputs explains the development of some well-known human perceptual preferences across different modalities.

2 Prioritized representation learning (PRL)

Let $\mathcal{R}$ be a representation model with parameters θ and let $\mathcal{L}$ be a loss function used to train this model with a dataset of inputs $\mathbf{X} = \{\mathbf{x}_i\}$. The representation model learns to map the inputs $\mathbf{x}_i$ into latent encodings $\mathbf{z}_i$. The loss measures the quality of these representations given the inputs, typically through a reconstruction [8] or contrastive [7] error. The parameters θ are updated by minimizing the loss computed over individual batches of inputs using gradient descent.

To capture the effect of active representation learning in the absence of control over the input sampling, we define a priority weighting function $\mathcal{W}(\mathbf{x}_i|\theta)$ that quantifies an "as-if active" preference for each input. $\mathcal{W}$ can be any arbitrary function representing an intrinsic preference for certain stimuli. Below we focus on compression as the intrinsic preference. In Appendix B.5 we comment on how different categories of intrinsic motivations [9] can be used to derive priority functions. Using the priority weighting function $\mathcal{W}$, we define the weighted loss function $\mathcal{L}^*$, given by:

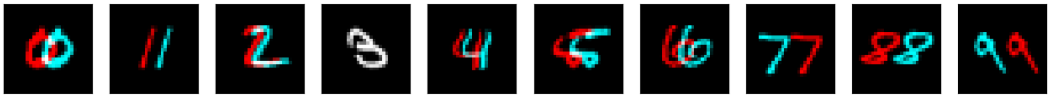
$$\mathcal{L}^* = \frac{\sum_i \mathcal{W}(\mathbf{x}_i|\bar{\theta})\; \mathcal{L}(\mathbf{x}_i|\theta)}{\sum_j \mathcal{W}(\mathbf{x}_j|\bar{\theta})}, \tag{1}$$

where the sums range over all elements in a batch. This weighted loss can be used instead of the original loss to update the parameters of the representation model. Notice that, if the representation model itself is used to compute the weights, its parameters must be frozen to prevent the flow of the backpropagation through the priority function, as indicated by the symbol $\bar{\theta}$. From Eq. (1), the terms with high priority weight, i.e. the terms corresponding to inputs that would be expected to be preferred in active representation learning, have higher contributions to the weighted loss function and thus should result in larger updates of the representation model. Algorithm 1 summarizes the method.

3 Experiments: "As-if Active" Efficient Coding

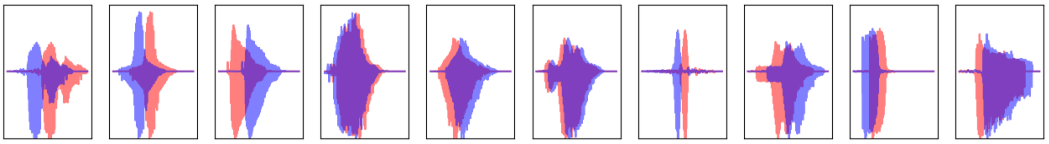
We demonstrate how PRL with a generic compression objective explains common human perceptual biases. Extensive experimental [10, 11], and computational [12, 13, 14] research points to coding

a) Binocular disparity



b) Symmetry

c) Interaural time difference



d) Exposure onset

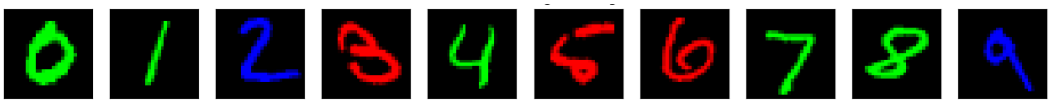


Figure 2: Examples of training data used for our experiments. The colors in the binocular disparity and interaural time difference datasets are only added for visualization, with the two channels being displayed as red and blue respectively. The colors in the exposure onset dataset are the actual colors of the RGB inputs.

efficiency as a strong indicator for preference, also closely related to the idea perceptual fluency [4, 1]. This principle is exploited in an active representation learning setting with the Active Efficient Coding (AEC) framework [15, 16], where agents learn to self-calibrate active perception systems with an intrinsic motivation to maximize coding efficiency. Our experiments show that PRL gives rise to similar perceptual biases as AEC but in a passive learning setting. Hence we also refer to this as "as-if Active" Efficient Coding (see Fig. 1). Below we give an overview of the experiments. Additional details are provided in Appendix C.

3.1 Datasets

All experiments make use of calibrated data, either from the MNIST image dataset [17] or the audio MNIST dataset [18]. We apply additional manipulations that vary **binocular disparity**, **symmetry**, **interaural time difference** (ITD), and **exposure onset**. Examples of each dataset with random manipulations are shown in Fig. 2. We ensure that the amount of information in the inputs remains constant for all manipulations, although redundancies for some inputs but not others may facilitate compression, and hence lead to a preference. From behavioral experiments with humans and non-human animals, we consider that the preferred states for each of these are, respectively, zero binocular disparity [19], symmetric views [1, 20], zero ITD [21], and familiar stimuli [22]. During training, the manipulations are sampled uniformly, such that each training batch contains a combination of inputs with different magnitudes of these manipulations.

3.2 Models

All models are deep autoencoders that compress the inputs of each dataset into latent encoding vectors. We define the loss function for a single input as a squared reconstruction error with an additional L1-regularization to induce sparsity [23, 24]:

$$\mathcal{L}(\mathbf{x}|\theta) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2 + \eta\|\theta(\mathbf{x})\|_1 = \frac{1}{D}\sum_{i=1}^{D}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2 + \frac{\eta}{M}\sum_{m=1}^{M}|\theta^{(m)}(\mathbf{x})|\,, \tag{2}$$

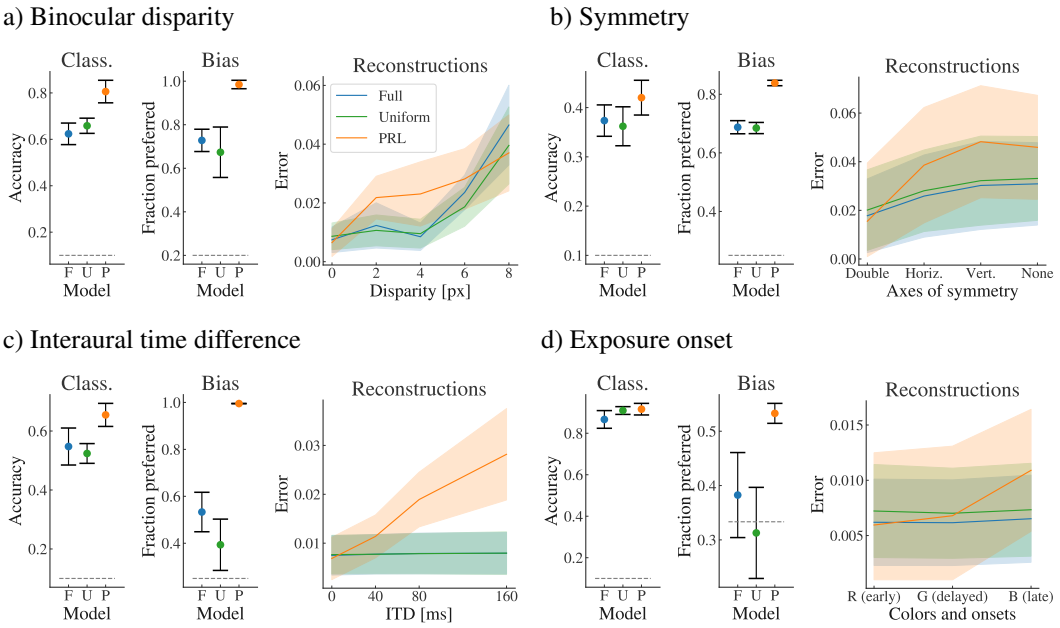


Figure 3: classification accuracies (left), bias preferences (center), and reconstruction errors (left) for the four experiments with different binocular disparities, symmetries, ITDs, and exposure onsets. Blue, green and orange colors are used for full (F), uniform (U), and PRL (P) models, respectively. Gray dashed lines indicate chance levels.

where the first sum ranges over all coordinates of the input of dimension D, the second term is the mean absolute activation of all the M hidden units of the autoencoder, and η is a scaling factor for sparsity. In our experiments we use $\eta = 0.01$.

The single-input loss from Eq. (2) is used to compute the weighted loss function for each training batch with a priority function reflecting the compressibility of each input given the current parameters of the autoencoder. More concretely, we measure compression as the quality of the representation standardized with the total information in the input. We approximate these two quantities with the negative reconstruction error of the autoencoder and the energy of the input, respectively (see Appendix B.5 and [15]), and use their ratio to define the priority function:

$$\mathcal{W}(\mathbf{x}|\bar{\theta}) = a_{k,\epsilon}\left[-\frac{\|\mathbf{x}-\hat{\mathbf{x}}\|_2}{E(\mathbf{x})}\right], \tag{3}$$

where the energy function E computes the absolute sum of a Laplacian filtered version of the images for the disparity, symmetry, and onset experiments, and the absolute sum of the magnitude of the sound wave for the ITD experiment. The activation function $a_{k,\epsilon}$ is a stochastic k-winners-take-all that converts the priority weights of the best $k(1-\epsilon)$ inputs of a batch, plus another $k\epsilon$ selected randomly, to 1 and the remaining ones to 0. By default we use $k = 50$ for a batch size of 250. This priority function guarantees that the model's resources are spent on further improving the representations of inputs with high compression by preventing inputs with low compression to contribute to the loss. A stochasticity parameter $\epsilon = 0.04$ is included, resulting in 2 of the 50 prioritized inputs being selected uniformly at random (see Appendix B.3). Alternative activations are discussed in Appendix B.2.

3.3 Results

After training the models on their respective datasets, we evaluate the learned representations on unseen test data to find human-like biases. Each input is given to the model with all possible manipulations and the reconstruction errors are recorded as a proxy for preference. We quantify the perceptual bias as the fraction of inputs which have lowest reconstruction error in the expected manipulation, e.g., the fraction of binocular inputs where the zero-disparity state is preferred. We also train linear classifiers to predict the labels from the representations of those preferred states,

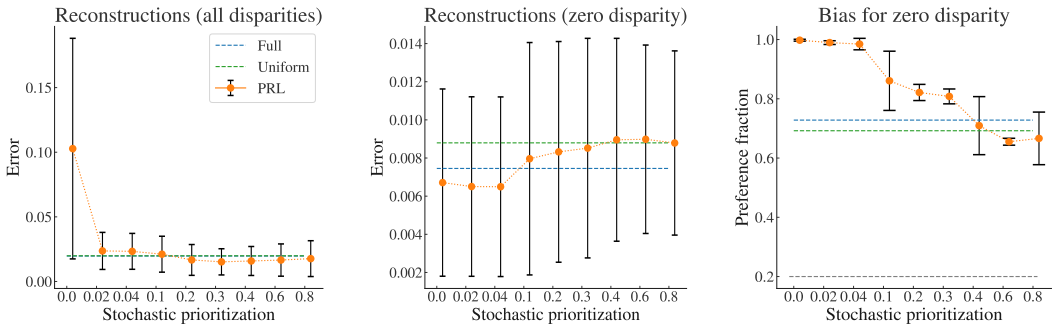


Figure 4: Analysis of the stochastic prioritization hyperparameter, ϵ. Plots show the average reconstruction error for all disparities (left) and zero disparity (center) states, and the bias preferences for zero disparity (right), for PRL models with increasing stochasticity in the k-winners-take-all activation. Blue and green dashed lines indicate the average values for the full and uniform models, respectively.

and evaluate them on the unseen test data. We compare PRL against two baselines trained without a priority weighted loss: first, a **uniform** sample of k inputs from the batch; second, a standard approach with **full** batches.

The main results are shown in Fig. 3. In all cases, we find that the models trained with PRL achieve marginally better reconstruction errors for the preferred states relative to the uniform and full models, but significantly higher perceptual biases and classification accuracies. However, for other manipulation magnitudes, the reconstruction errors are higher, i.e. worse, when using PRL. This is consistent with human representations [19, 1, 21] and highlights how many sensory biases can be explained by an (as-if) active preference for compression.

To better understand the performance of PRL, we explore the effects of the stochasticity parameter. Fig. 4 shows the reconstruction error and bias for the binocular disparity experiment. Using $\epsilon = 0$ results in high overall reconstruction errors, while using a high stochasticity makes the bias for the human preference disappear. From this analysis, we select $\epsilon = 0.04$ as the default stochasticity for our PRL models. Additional results are provided in Appendix C.5. We show that these human-like biases originate in correlations between the magnitude of the manipulations and the priorities within a batch. That is, we find that our method amplifies biases by selectively training on the preferred inputs, since these have higher compression and hence contribute more to the PRL weighted loss function.

4 Conclusions

We have presented prioritized representation learning (PRL), a model of the formation of common human perceptual preferences. PRL introduces a priority weighted loss function inspired by active representation learning. This allows PRL to filter out uninteresting inputs in order to save model resources and improve the representations of more interesting ones. PRL can therefore be linked to human attentional mechanisms [25, 26]. We show that a priority function favoring compression of sensory inputs captures common perceptual biases across different sensory modalities (vision: small binocular disparities, audition: low interaural time differences), as well as aesthetic preferences for symmetries and familiarity. The formalism and much of the procedure presented here can be readily adapted to other priorities. Future work should explore the representations learned with alternative priority functions, as well as the incorporation of PRL into active learning settings.

Code availability The code of the experiments is available at https://github.com/trieschlab/as-if-aec.

Acknowledgements This research was supported in part by "The Adaptive Mind" funded by the Hessian Ministry of Higher Education, Research, Science and the Arts, Germany, by the German Research Foundation (DFG) Project numbers 520617944, 520223571 ("Sensing LOOPS"), and by the Hong Kong Research Grant Council (Grant 16213322). JT was supported by the Johanna Quandt foundation.

References

[1] Rolf Reber, Norbert Schwarz, and Piotr Winkielman. Processing fluency and aesthetic pleasure: Is beauty in the perceiver's processing experience? *Personality and social psychology review*, 8(4):364–382, 2004.

[2] Tiffany M Field, Debra Cohen, Robert Garcia, and Reena Greenberg. Mother-stranger face discrimination by the newborn. *Infant Behavior and development*, 7(1):19–25, 1984.

[3] Alison K Ventura and John Worobey. Early influences on the development of food preferences. *Current biology*, 23(9):R401–R408, 2013.

[4] Rolf Reber, Piotr Winkielman, and Norbert Schwarz. Effects of perceptual fluency on affective judgments. *Psychological science*, 9(1):45–48, 1998.

[5] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2016.

[6] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[8] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993.

[9] Arthur Aubret, Laetitia Matignon, and Salima Hassas. An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey. *Entropy*, 25(2):327, 2023.

[10] Michael S Lewicki. Efficient coding of natural sounds. *Nature neuroscience*, 5(4):356–363, 2002.

[11] Rafael Polanía, Michael Woodford, and Christian C Ruff. Efficient coding of subjective value. *Nature neuroscience*, 22(1):134–142, 2019.

[12] Bruno A Olshausen and David J Field. Natural image statistics and efficient coding. *Network: computation in neural systems*, 7(2):333, 1996.

[13] Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

[14] Arthur Aubret, Mathieu Lefort, Céline Teulière, Laetitia Matignon, Salima Hassas, and Jochen Triesch. Compressed information is all you need: unifying intrinsic motivations and representation learning. In *NeurIPS 2022 Workshop on Information-Theoretic Principles in Cognitive Systems*, 2022.

[15] Samuel Eckmann, Lukas Klimmasch, Bertram E Shi, and Jochen Triesch. Active efficient coding explains the development of binocular vision and its failure in amblyopia. *Proceedings of the National Academy of Sciences*, 117(11):6156–6162, 2020.

[16] Francisco M López, Marcel C Raabe, Bertram E Shi, and Jochen Triesch. Self-calibrating saccade-vergence interactions. In *2024 IEEE International Conference on Development and Learning (ICDL)*, pages 1–7. IEEE, 2024.

[17] Yann LeCun. The mnist database of handwritten digits. *http://yann.lecun.com/exdb/mnist/*, 1998.

[18] Sören Becker, Johanna Vielhaben, Marcel Ackermann, Klaus-Robert Müller, Sebastian Lapuschkin, and Wojciech Samek. Audiomnist: Exploring explainable artificial intelligence for audio analysis on a simple benchmark. *Journal of the Franklin Institute*, 361(1):418–428, 2024.

[19] Ian P Howard and Brian J Rogers. *Binocular vision and stereopsis*. Oxford University Press, USA, 1995.

[20] Andrew Isaac Meso, Anna Montagnini, Jason Bell, and Guillaume S Masson. Looking for symmetry: Fixational eye movements are biased by image mirror symmetry. *Journal of Neurophysiology*, 116(3):1250–1260, 2016.

[21] G Linn Roth, Ravindra K Kochhar, and Joseph E Hind. Interaural time differences: implications regarding the neurophysiology of sound localization. *The Journal of the Acoustical Society of America*, 68(6):1643–1651, 1980.

[22] Katharina Riebel. Early exposure leads to repeatable preferences for male song in female zebra finches. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 267(1461):2553–2558, 2000.

[23] Devansh Arpit, Yingbo Zhou, Hung Ngo, and Venu Govindaraju. Why regularized auto-encoders learn sparse representation? In *International Conference on Machine Learning*, pages 136–144. PMLR, 2016.

[24] Li Zhang, Yaping Lu, Bangjun Wang, Fanzhang Li, and Zhao Zhang. Sparse auto-encoder with smoothed l 1 regularization. *Neural Processing Letters*, 47(3):829–839, 2018.

[25] Daniel J Simons and Christopher F Chabris. Gorillas in our midst: Sustained inattentional blindness for dynamic events. *perception*, 28(9):1059–1074, 1999.

[26] Angela Radulescu, Yeon Soon Shin, and Yael Niv. Human representation learning. *Annual Review of Neuroscience*, 44(1):253–273, 2021.

[27] Hu Li, Xuezhong Qian, and Wei Song. Prioritized experience replay based on dynamics priority. *Scientific Reports*, 14(1):6014, 2024.

[28] K Ruwani M Fernando and Chris P Tsokos. Dynamically weighted balanced loss: class imbalanced learning and confidence calibration of deep neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(7):2940–2951, 2021.

[29] Jonathon Byrd and Zachary Lipton. What is the effect of importance weighting in deep learning? In *International conference on machine learning*, pages 872–881. PMLR, 2019.

[30] Mahsa Salehi, Xuyun Zhang, James C Bezdek, and Christopher Leckie. Smart sampling: A novel unsupervised boosting approach for outlier detection. In *AI 2016: Advances in Artificial Intelligence: 29th Australasian Joint Conference, Hobart, TAS, Australia, December 5-8, 2016, Proceedings 29*, pages 469–481. Springer, 2016.

[31] Robert E Schapire. The boosting approach to machine learning: An overview. *Nonlinear estimation and classification*, pages 149–171, 2003.

[32] Arun Suggala, Bingbin Liu, and Pradeep Ravikumar. Generalized boosting. *Advances in neural information processing systems*, 33:8787–8797, 2020.

[33] Richard Held and Alan Hein. Movement-produced stimulation in the development of visually guided behavior. *Journal of comparative and physiological psychology*, 56(5):872, 1963.

[34] Nate Kornell, Lisa K Son, and Herbert S Terrace. Transfer of metacognitive skills and hint seeking in monkeys. *Psychological Science*, 18(1):64–71, 2007.

[35] Oliver Baumann, Edgar Chan, and Jason B Mattingley. Dissociable neural circuits for encoding and retrieval of object locations during active navigation in humans. *Neuroimage*, 49(3):2816–2825, 2010.

[36] KH James, GK Humphrey, T Vilis, B Corrie, R Baddour, and MA Goodale. "active" and "passive" learning of three-dimensional object structure within an immersive virtual reality environment. *Behavior Research Methods, Instruments, & Computers*, 34(3):383–390, 2002.

[37] Robert A Bjork, John Dunlosky, and Nate Kornell. Self-regulated learning: Beliefs, techniques, and illusions. *Annual review of psychology*, 64(1):417–444, 2013.

[38] Arthur Aubret, Céline Teulière, and Jochen Triesch. Self-supervised visual learning from interactions with objects. *arXiv preprint arXiv:2407.06704*, 2024.

[39] Shuhan Tan, Tushar Nagarajan, and Kristen Grauman. Egodistill: Egocentric head motion distillation for efficient video understanding. *Advances in Neural Information Processing Systems*, 36:33485–33498, 2023.

[40] Yifei Wang, Kaiwen Hu, Sharut Gupta, Ziyu Ye, Yisen Wang, and Stefanie Jegelka. Understanding the role of equivariance in self-supervised learning. In *ICML 2024 Workshop on Theoretical Foundations of Foundation Models*, 2024.

[41] Jongmin Lee, Byungjin Kim, Seungwook Kim, and Minsu Cho. Learning rotation-equivariant features for visual correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21887–21897, 2023.

[42] Alexandre Devillers and Mathieu Lefort. Equimod: An equivariance module to improve visual instance discrimination. In *International Conference on Learning Representations*, 2023.

[43] Xia Xu and Jochen Triesch. Ciper: Combining invariant and equivariant representations using contrastive and predictive learning. In *International Conference on Artificial Neural Networks*, pages 320–331. Springer, 2023.

[44] Adam Stooke, Kimin Lee, Pieter Abbeel, and Michael Laskin. Decoupling representation learning from reinforcement learning. In *International conference on machine learning*, pages 9870–9879. PMLR, 2021.

[45] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.

[46] Riashat Islam, Raihan Seraj, Pierre-Luc Bacon, and Doina Precup. Entropy regularization with discounted future state distribution in policy gradient methods. *arXiv preprint arXiv:1912.05104*, 2019.

[47] Vitchyr H Pong, Murtaza Dalal, Steven Lin, Ashvin Nair, Shikhar Bahl, and Sergey Levine. Skew-fit: State-covering self-supervised reinforcement learning. *arXiv preprint arXiv:1903.03698*, 2019.

[48] Giulia Vezzani, Abhishek Gupta, Lorenzo Natale, and Pieter Abbeel. Learning latent state representation for speeding up exploration. *arXiv preprint arXiv:1905.12621*, 2019.

[49] Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A Efros. Large-scale study of curiosity-driven learning. *arXiv preprint arXiv:1808.04355*, 2018.

[50] Bradly C Stadie, Sergey Levine, and Pieter Abbeel. Incentivizing exploration in reinforcement learning with deep predictive models. *arXiv preprint arXiv:1507.00814*, 2015.

[51] Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pages 2721–2730. PMLR, 2017.

[52] Marlos C Machado, Marc G Bellemare, and Michael Bowling. Count-based exploration with the successor representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5125–5133, 2020.

[53] Arthur Aubret, Laetitia Matignon, and Salima Hassas. Distop: Discovering a topological representation to learn diverse and rewarding skills. *IEEE Transactions on Cognitive and Developmental Systems*, 15(4):1905–1915, 2023.

[54] Lakshitha P Wijesinghe, Melville J Wohlgemuth, Richard HY So, Jochen Triesch, Cynthia F Moss, and Bertram E Shi. Active head rolls enhance sonar-based auditory localization performance. *PLoS computational biology*, 17(5):e1008973, 2021.

[55] Francisco M López, Bertram E Shi, and Jochen Triesch. Eye-hand coordination develops from active multimodal compression. In *2023 IEEE International Conference on Development and Learning (ICDL)*, pages 437–442. IEEE, 2023.

[56] Yu-Lin He, Xiao-Liang Zhang, Wei Ao, and Joshua Zhexue Huang. Determining the optimal temperature parameter for softmax function in reinforcement learning. *Applied Soft Computing*, 70:80–85, 2018.

[57] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.

[58] Hamed Sarvari, Carlotta Domeniconi, Bardh Prenkaj, and Giovanni Stilo. Unsupervised boosting-based autoencoder ensembles for outlier detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 91–103. Springer, 2021.

[59] B Ravi Kiran, Dilip Mathew Thomas, and Ranjith Parakkal. An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos. *Journal of Imaging*, 4(2):36, 2018.

[60] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[61] Daniel Y Little and Friedrich T Sommer. Learning and exploration in action-perception loops. *Frontiers in neural circuits*, 7:37, 2013.

[62] Céline Craye, Timothée Lesort, David Filliat, and Jean-François Goudou. Exploring to learn visual saliency: The rl-iac approach. *Robotics and Autonomous Systems*, 112:244–259, 2019.

[63] Xinchen Lin, Yang Tang, Huaglory Tianfield, Feng Qian, and Weimin Zhong. A novel approach to reconstruction based saliency detection via convolutional neural network stacked with auto-encoder. *Neurocomputing*, 349:145–155, 2019.

[64] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

[65] Andrew Levy, George Konidaris, Robert Platt, and Kate Saenko. Learning multi-level hierarchies with hindsight. *arXiv preprint arXiv:1712.00948*, 2017.

[66] Felix Schneider, Xia Xu, Markus R Ernst, Zhengyang Yu, and Jochen Triesch. Contrastive learning through time. In *SVRHM 2021 Workshop@ NeurIPS*, 2021.

[67] Xin Yuan, Zhe Lin, Jason Kuen, Jianming Zhang, Yilin Wang, Michael Maire, Ajinkya Kale, and Baldo Faieta. Multimodal contrastive training for visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6995–7004, 2021.

[68] Gillian Rhodes, Fiona Proffitt, Jonathon M Grady, and Alex Sumich. Facial symmetry and the perception of beauty. *Psychonomic Bulletin & Review*, 5:659–669, 1998.

[69] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *SciPy*, pages 18–24, 2015.

[70] Robert B Zajonc. Mere exposure: A gateway to the subliminal. *Current directions in psychological science*, 10(6):224–228, 2001.

[71] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

A Related works

PRL uses a priority weighted loss function, where the priorities are computed over each batch sampled from the dataset. This approach is reminiscent of a number of different machine learning methods, detailed below.

A.1 Non-uniform sampling and weighted losses

As mentioned in the Introduction, this work draws inspiration from prioritized experience replay (PER) [5] in reinforcement learning. PER introduces non-uniform sampling from an experience replay buffer based on the rewards collected by the agent. Concretely, state-action-state transitions are weighted by

$$P(i) = \frac{p_i^\alpha}{\sum_j p_j^\alpha} \tag{4}$$

where $p_i = |\delta_i| + \epsilon$ and δ_i is the temporal-difference error that can quantify the learning made during that transition. ϵ is added as a stochastic prioritization. In PER, more relevant (i.e. highly informative) transitions are used more to train the agent, which optimizes the learning. In PRL, we take inspiration from this approach to likewise positively weight inputs that are deemed more relevant given the priority function of choice. Extensions of PER show further improvements to this method, e.g. by dynamically adjusting the priority weights [27] to accelerate convergence.

On the other hand, since PRL introduces priorities in a weighted loss, it is also relevant to comment on the use of such weighted losses in unsupervised learning. This approach is most frequently used to deal with class imbalance [28], where under-represented inputs in a dataset need to be positively weighted. Additionally, weighted losses can accelerate training in the early training stages [29] and enhance downstream tasks such as outlier detection [30]. Weighted losses are also common in boosting techniques is that aim to minimize classification errors by training weak models to create a strong classifier [31, 32]. The predictions of the weak models are combined sequentially and weighted, so that further training can be performed on inputs with low accuracies. In particular, if a single weak model is used, then boosting is equivalent to using a loss that positively weights inputs with poor classification [31]. Overall, the methods mentioned here share a common goal of trying to improve performance in the face of unbalanced or difficult data. While they share some common ground with our work, these approaches cannot capture perceptual preferences in the same way that PRL can.

A.2 Actively learned representations

In the 1960s, Held and Hein showed that kittens who could voluntarily move in their environments during early stages of their visual development learned better depth perception [33]. Similar benefits for active learning are ubiquitous for humans and non-human animals across different modalities and behaviors, including visual representations [34], navigation [35], hand-held object recognition[36], and more [37]. However, for most of these behavioral experiments, it is difficult if not impossible to determine exactly what the contribution of active learning is: is it merely that subjects learn better representations because they can sample their sensory inputs voluntarily, or are they somehow also encoding actions and their consequences? Recent approaches in self-supervised learning explore the latter idea by learning joint representations for actions and sensory perceptions [38, 39] or by learning equivariant representations from the consequences of those actions [40, 41, 42, 43]. However, these aim to learn representations over an entire dataset and lack any control over input sampling.

On the other hand, the idea of active representation learning is also of key importance in the field of reinforcement learning [44]. Intrinsic motivations can be particularly sensitive to representations, since many rely on state representations and their distributions through active exploration [45, 46, 47, 48], information maximization [49, 50], pseudocounts [51, 52], efficient coding [15, 16], and other representation-dependent metrics. A common denominator for these works is that the representations are typically learned in parallel to the behaviors. That is, as an agent learns its policy, it also needs to improve its representation model, e.g. updating the pseudocounts estimations [52]. A considerable advantage of this concurrent learning of representations and behaviors is that the (typically limited) resources of the representational model can be re-assigned to those states that are more useful given the current policy of the agent. And since this policy is driven by the representations, the result is a positive feedback loop optimized for exploration [52], skill-learning [53], or coordination [15].

B Formalism

B.1 A priority for compression

In this work we focus on efficient coding as a priority for compression. The idea of actively maximizing coding efficiency as an intrinsic motivation is not new [15, 16, 54]. An efficient code maximizes the mutual information between inputs $\mathbf{x}$ and their representations $\mathbf{z}$. Fundamentally, an agent driven by an efficient coding motivation will aim to maximize the marginal entropy of its inputs (exploration) while minimizing the conditional entropy of the input given the latent representation (reconstruction) [15]:

$$MI(\mathbf{x}, \mathbf{z}) = \mathcal{H}(\mathbf{x}) - \mathcal{H}(\mathbf{x}|\mathbf{z}). \tag{5}$$

To do so, an an agent should try to receive inputs with high entropy and represent them with as little loss of information as possible. The two terms can be attributed to different behaviors. In the context of binocular vision, [15] maximize the marginal entropy with accommodation actions that increase the contrast of the image, while the conditional entropy is minimized with vergence actions that reduce binocular disparity. Likewise, [16] also minimize the conditional entropy with vergences but the marginal entropy of the inputs is maximized with saccades.

The experiments presented in this paper follow a similar approach. In line with [15], we approximate the two terms of Eq. (5) with the total energy of an input and its representation loss, respectively (see Appendix C.2). However, using the difference of the two values as a priority poses a problem: as the representation quality improves, the conditional entropy term decreases and the priorities are only determined by the energy of the inputs. This is not in the spirit of PRL. [16] also identify that the development of representations introduce instabilities, and therefore use the relative change in the reconstruction error of an autoencoder between two consecutive states as an intrinsically-motivated reward; they find that this accelerates learning with respect to alternative rewards. This is not possible with passive representations, since inputs are treated in isolation rather than as transitions between two states.

We propose an alternative solution inspired by [16], whereby the representation loss is normalized by the energy of the input. Intuitively, inputs with higher energy have more information to encode, and thus should have higher representation errors. However, models that can learn to exploit redundancies in the encoding process [55] can also achieve high-quality representations for inputs with high energy.

In doing so, we define compression as the quality of the representation divided by the energy of the input, given for example by:

$$w(\mathbf{x}|\theta) = -\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|}{E(\mathbf{x})}, \tag{6}$$

where $\hat{\mathbf{x}}$ is the reconstruction of the input by an autoencoder and the minus sign is used because better representations have lower reconstruction errors. While this can in principle be used directly to compute the relative preferences for different inputs, it is often beneficial to further segregate inputs with high and low compressions by means of an activation function (see below). Thus, the compression priority function shown in Eq. (3.2) becomes $\mathcal{W}(\mathbf{x}|\theta) = a[w(\mathbf{x}|\theta)]$.

Let us make one last comment with respect to the definition of compression given by Eq. 6. In principle, one may be tempted to think that compression should be an intrinsic property of the inputs. For example, as mentioned previously, inputs with redundancies should be more compressible, i.e. have higher *compressibility*. This is the case for many perceptual properties. Out of the four experiments presented in this work, three of them refer to universal properties that originate in the exploit of such redundancies: binocular disparity and ITD exploit redundancies in binocular vision and binaural hearing, respectively, while symmetry exploits redundancies within a single image. However, in the exposure onset experiment, there are no such redundancies. It is entirely arbitrary which particular inputs are preferred by the agent, since the bias only originates from the earlier exposure. This demonstrates that it is the compression of the inputs given the current state of the representation model, and not an intrinsic compressibility of the inputs *a priori*, that must be at the basis of the development of these perceptual preferences.

B.2 Activation functions

Priority functions can be directly used to computed a weighted loss function. However, in most practical cases, additional magnifications are required. By this, we mean that activation functions can be applied on top of the priority functions to increase or decrease the weighting values. Here we discuss three cases.

First, since unsupervised learning methods rely on batches of fixed length, it is reasonable to select an activation function that changes the size of the batches. This is the case of the k-winners-take-all used in the experiments of this paper. Hence, rather than using the entire batch of size N, a sub-batch of size k is used to train the representation model. Since these are the inputs with k highest priorities, training focuses on the most relevant inputs.

An alternative to using the inputs with the k highest priorities of a batch is to define a priority threshold and train on all inputs of a batch exceeding this threshold. Since the priorities are expected to change during training, the use of an adaptive threshold may be preferable.

Finally, a common procedure is to use a Softmax activation function $\sigma_\tau(\cdot)$, where the temperature hyperparameter τ can be tuned to control the distribution of weights. The choice of temperature in the Softmax has considerable consequences in the resulting weighted loss (see [56]). From exploratory experiments (not shown) we find that the best approach may be to control the temperature hyperparameter during training, e.g. to maintain a constant entropy of the batches.

B.3 Stochastic prioritization

The priority function ensures that some inputs provide higher contributions to the loss function and thus the representation model is optimized for those inputs. A possible downside of this method is that the representations of non-prioritized inputs may be too poor, making them virtually unusable for downstream tasks. In PER [5], the authors introduce a stochastic prioritization term, which fundamentally consists of adding a small value ϵ to all transitions' priorities, increasing the chances of sampling otherwise ignored transitions.

We follow a similar idea here by adding stochasticity to the priorities within a batch. The specific way in which this is done depends on the activation function used. For example, for a Softmax activation, the addition of random values to the weights can increase the priorities of ignored inputs. When using a k-winners-take-all activation, as done in the experiments shown here, the stochastic prioritization can instead be the fraction of those k inputs that are selected randomly from the batch, rather than using the priority function. More specifically, we define a stochasticity parameter $\epsilon \in [0, 1]$ such that

the activation function first selects the $k(1-\epsilon)$ inputs with highest priority, and another $k\epsilon$ ones are sampled uniformly from the $N - k(1-\epsilon)$ inputs remaining on the batch. By default we use $\epsilon = 0.04$ in all experiments. In Appendix C.5.4 we evaluate representations learned with increasing values of ϵ.

B.4 PRL approximates active representation learning

Our method is largely inspired by active learning and its role in shaping representations. PRL can prevent representational specificities from being lost when using passive approaches, i.e. supervised, self-supervised, and unsupervised learning. Since the standard implementations of these methods lack any kind of control over their input statistics, and instead these are sampled uniformly from a predefined training dataset, the overall goal for passive approaches is to achieve better representations over an entire dataset. This is rarely the case in active settings: states which are not interesting stop being visited, and learning accurate representations for them becomes useless. Thus, at a coarse level, PRL can be interpreted as a filtering mechanism for irrelevant inputs within a training dataset.

However, going beyond this coarse level, we find that PRL can be a testbed for new active learning methods. The experiments shown in this paper can serve as an example. While implementing AEC experiments requires creating an environment, defining an agent with its observation and action spaces, and using reinforcement learning to train the agent's policy, our as-if AEC approach only involves unsupervised training on a fixed dataset – and yet the results achieved are comparable. Admittedly it is not possible to evaluate behaviors without a trained agent, but nonetheless PRL can facilitate the initial exploratory stages and proof-of-concept validations for innovative approaches.

B.5 From intrinsic motivations to priority functions

PRL requires a priority weighting function, but it does not provide any indications as to which function to choose. Some specific experiments may benefit from the use of task-specific functions, e.g. trying to enhance representations required for some external goal-directed behavior. Such cases can incorporate extrinsic or supervised metrics that can be immediately converted into priority functions. We will not comment on these any further. Instead, here we want to give some initial insights into how different intrinsic motivations can be converted into priorities for open-ended unsupervised learning. See [9] for a detailed review and examples of intrinsic motivations.

B.5.1 Novelty

Novelty is a common intrinsic motivation that promotes exploration. Seeking novelty means seeking states which have not been visited before. In very large or continuous spaces, keeping track of previously visited states is difficult if not impossible [57]. A more manageable approach is to rely on pseudocounts within the learned representation space [51, 52]. Novelty can be interpreted as an input being out-of-distribution relative to a previously trained-on dataset. That is, novel states in a reinforcement learning framework can correspond to outliers in an unsupervised learning framework. Thus, the literature on anomaly detection can be of use [58, 30, 59] for defining appropriate priorites.

There are multiple priority functions that can be derived from a novelty intrinsic motivation. First, using the distance of each input to all other inputs within a batch, one can prioritize inputs that are most different. Likewise, using an additional buffer memory that compares inputs from multiple batches, the procedure can be extended to the entire dataset. More interesting priority functions can rely on the learned representations themselves, e.g. prioritizing inputs with high reconstruction errors, $\mathcal{W}(\mathbf{x}|\theta) = \|\mathbf{x} - \hat{\mathbf{x}}\|$. Maximizing the entropy of the visited states is also possible using the loss of a variational autoencoder [60, 48]:

$$\log \rho(\mathbf{x}) \approx -\log q_{decoder}(\mathbf{x}|\mathbf{z}) + D_{KL}\left(q_{encoder}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})\right), \tag{7}$$

where $\rho(\mathbf{x})$ is the probability distribution over visited states, $q_{encoder}(\cdot)$ and $q_{decoder}(\cdot)$ are the encoder and decoder networks, D_{KL} is the Kullback–Leibler divergence, and $\mathrm{p}(\mathbf{z})$ is the prior distribution over latent vectors (see [9] for details). [48] use Eq. (7) to reward actions that move the agent towards states that maximize this loss. In the context of PRL, it can also be used directly as a priority function, i.e. $\mathcal{W}(\mathbf{x}|\theta) = -\log \rho(\mathbf{x})$.. Interestingly, such a priority function should have the opposite effect as the compression priority used in our experiments: accurate representations are neglected in favor of novel ones. Thus, novelty-based priorities for unsupervised learning may have similar results as boosting methods (see Appendix A).

B.5.2 Surprise

Unlike novelty, surprise-based rewards are mainly associated with state transitions. That is, a state-action-state transition (s, a, s') transition is surprising if the new state is in any way unexpected given the previous state and the action taken. Mathematically, surprise may be defined as the distance between the new state and a prediction from a forward model $\hat{s}'$. This distance is commonly measured in a representation space rather than the input space, , i.e. $d = \|f(s') - f(\hat{s}')\|$, where the function $f(\cdot)$ can be, for example, a representation encoder [50]. Some improvements to this basic approach are possible, e.g. rewarding the improvement of the predictions of the forward model [61]. Regardless of the intrinsic motivation, surprise always considers the transition to a new state. It is immediate to see that such methods can not be accurately approximated with PRL, since only single inputs are sampled randomly from a dataset.

Nevertheless, momentarily stepping away from intrinsic motivations, there is an alternative interpretation of surprise that may be of use in PRL. Sensory inputs can be surprising by the context in which they are presented. For example, one would be rather surprised to find a lion on the streets of Berlin, but not in the African savanna. In this case, we suggest to measure surprise by the saliency of the input itself. That is, parts of an input can be surprising (salient) given the rest of the input. As far as we are aware, only few studies explore saliency as an intrinsic motivation and their success is moderate, e.g. [62]. On the other hand, the literature on saliency detection is vast and varied, and many of these models can serve as inspiration for priority functions. Of particular relevance here, and related to the idea of surprise as indicative of an input being unexpected in its context, [63] detect saliency using autoencoders that predict the contents of patches removed from an image. It remains to be seen whether the use of saliency as a priority in such a model can lead to representations that are optimized for saliency detection.

B.5.3 Skill-learning

Finally, many intrinsic motivations can be classed under the general umbrella of skill-learning. For example, maximizing compression can be interpreted as intrinsically-motivated skill-learning, insofar as it allows for the development of self-calibrated behaviors [14, 15, 16]. Likewise, compression can be at the source of multimodal and multi-effector coordination [55].

[9] define skill-learning as the maximization of the mutual information between goals and state trajectories. They propose a distinction between methods that learn to execute goals for different parts of the state space and methods that derive goal spaces from state spaces. The latter is not particularly relevant in the context of our work. The first methods typically try to minimize the distance between a state and its corresponding goal – the goals themselves also need to be appropriately discovered. Many of these methods rely on goal-conditioned policies [64, 65]. Likewise, using PRL it is possible to learn goal-conditioned representations. Here, of course, the term "goal" is used rather liberally, since there are no actions to be performed and thus no attainment possible. We use the term for key or anchor inputs within certain subcategories of a dataset. Using the distance between each input's representation $\mathbf{z}$ and that of its corresponding goal $\mathbf{z}_g$, we can propose the skill-learning priority $\mathcal{W}(\mathbf{x}|\theta) = \|\mathbf{z} - \mathbf{z}_g\|$. This priory is somewhat reminiscent of contrastive losses in self-supervised learning, e.g. [7], if we consider the goal $\mathbf{x}_g$ to be one input of a class and expect other inputs $\mathbf{x}$ to have similar representations. Thus, the use of this particular priority would focus the learning on those inputs with dissimilar representations to that of their goals. As for the selection of goal inputs, these can simply be other inputs with artificial [7], temporal [66], or multimodal [67] augmentations. Alternatively goal representations can be directly sampled from the latent space despite not corresponding to any particular input, e.g. by selecting cluster centers.

C Experiments

C.1 Datasets and models

Our experiments use calibrated datasets [17, 18] with additional manipulations. Here we detail each of them and provide insights into the human preferences based on behavioral experiments. Examples of the datasets and manipulations are shown in Fig. 2. We also describe the autoencoder used as representation model for each dataset.

C.1.1 Binocular disparity

When looking at a scene, humans align both eyes on an object to achieve a fused single representation [19, 15], a configuration referred to as zero-disparity, whereas higher binocular disparities can produce double images. This behavior has been extensively studied and reproduced in the AEC framework [15, 55, 16]. For example, [16] use deep convolutional autoencoders to learn abstract representations of binocular inputs within an active sensorimotor loop (see Fig. 1), and find a global minimum of the reconstruction error at the zero-disparity state. This perceptual preference results in a learned vergence behavior that aligns the two eyes on a target. Here, we aim to reproduce these results with PRL. We explore the visual bias for zero-disparity states by augmenting a stereoscopic version of the MNIST dataset with random binocular disparities, i.e. horizontal shifts in opposite directions for the two channels (left and right). The original MNIST images have a resolution of 28×28 pixels. To prevent high disparities from moving the images beyond their borders, we add a padding of 8 pixels on each side, then shift the images by magnitudes $\delta = \{0, \pm 2, \pm 4, \pm 6, \pm 8\}$ pixels, and then resize the images to 28×28 pixels.

The inputs to the autoencoder are therefore tensors of size $(2 \times 28 \times 28)$, which are encoded into a latent space of size (6). This is done by an encoder with three 2-dimensional convolutional layers with parameters[1] $(16, 3, 2, 1)$, $(32, 3, 2, 1)$, and $(64, 3, 2, 0)$, followed by a two fully connected layers of parameters[2] $(576, 256)$ and $(256, 6)$. Additionally we use ReLU activations after each layer except the last one and batch normalization after the convolutional layers. The decoder is a similar architecture, with two fully connected layers of parameters $(6, 256)$ and $(256, 576)$ followed by three 2-dimensional transpose convolutional layers of parameters[3] $(32, 3, 2, 0, 0)$, $(16, 3, 2, 1, 1)$, and $(2, 3, 2, 1, 1)$, with ReLU activations and batch normalization after the first two transpose convolutional layers.

C.1.2 Symmetry

Symmetric images are perceived as more aesthetically pleasing [1, 68] and symmetry axes are fixated more often when available [20]. Ongoing work (unpublished) shows that these preferences can emerge with AEC. In the experiment presented here, we evaluate the symmetry bias with increasingly symmetric versions of the MNIST images, i.e. with additional axes of reflectional symmetry through the center of the image. The manipulations consist of adding a horizontal symmetry, vertical symmetry, or both. During training, the half of the image that is retained for the reflectional symmetry is sampled randomly. When testing the model, we apply both reflections and take the average of their reconstruction errors and classification accuracies.

The autoencoder used for the symmetry experiment is similar to the one used for the binocular disparity experiments, with some minor differences since the symmetry inputs have a single channel, i.e. tensors of size $(1 \times 28 \times 28)$. The latent space in this case has a size of (5). For the encoder, the number of filters of the convolutional layers are, respectively, $8, 16, 32$, and the units of the fully connected layers are, respectively, $288, 128, 5$. Likewise, the same values are used for the decoder.

C.1.3 Interaural time difference (ITD)

Humans and non-human animals orient their heads in the direction of a sound source such as to minimize the ITD, i.e. the temporal delay in the arrival of auditory stimuli to both ears [21]. Previous work in the AEC framework explored the role of ITD in bat echolocation experiments [54]. There, a bat model aligned its head orientation with a sound source based on an intrinsic motivation to minimize the reconstruction error of binaural signals. The experiments showed that pointing the head to the source of the sound, resulting in zero ITD, achieves the lowest reconstruction error. Here, we create a binaural extension of the auditory MNIST dataset and randomly delay the phase of one of the two channels by $\delta = \{0, 40, 80, 160\}\ ms$. It should be noted that in real binaural scenarios there is an additional interaural level difference (ILD), whereby the amplitude of the two signals is different. This effect was also considered in [54] but we neglect it here for simplicity.

The audio time series are converted into the frequency domain by means of a mel-spectrogram [69]. Each audio is first resampled to a frequency of 12 kHz, clipped to a total duration of one second,

[1] Using the notation (*filters*, *filter size*, *stride*, *padding*) for convolutional layers

[2] Using the notation (*in units*, *out units*) for fully connected layers

[3] Using the notation (*filters*, *filter size*, *stride*, *padding*, *output padding*) for transpose convolutional layers

and divided into 28 partially-overlapping time windows. A mel-spectrogram is computed for each window with 28 frequency bands. The result is a spectrogram of dimensions $2 \times 28 \times 28$, ordered as binaural channel, frequency band, and time window. We build the spectrograms this way so that we can use the same autoencoder architecture of the binocular disparity experiments (see Appendix C.1.1).

C.1.4 Exposure onset

The last experiments explore the effects of exposure onset on perceptual preferences. Numerous phenomena point to the times and orders in which different categories of stimuli are discovered as a main source for perceptual preference. In particular, the mere-exposure effect is a well-established psychological effect that explains affect based on familiarity [70]. This effect is most notorious as the result of early experiences in childhood [22, 3], but can also develop on a shorter time scale as a consequence of a perceptual fluency for common or repeated stimuli [4]. In this experiment, we simulate the emergence of perceptual preferences based on exposure onsets using a color version of the MNIST image dataset, with red, green, or blue images. The training is divided into 3 stages: during the first 5 epochs, only red images are shown to all models; during the next 5 epochs, images are either red or green. For the remaining 20 epochs, all 3 colors are used. Therefore, during the sixth and eleventh epoch, the green and blue images respectively are completely new and present an interesting conundrum for our PRL model: the priority weighting function should filter out these new colors and continue updating its weights to accurately represent the familiar stimuli, i.e. the red images.

The model is similar to the ones described previously, but with additional filters and a larger bottleneck. The inputs have a dimensions $3 \times 28 \times 28$. The encoder has three 2-dimensional convolutional layers of parameters $(24, 3, 2, 1)$, $(48, 3, 2, 1)$, and $(96, 3, 2, 0)$, followed by a two fully connected layers of parameters $(864, 384)$ and $(384, 10)$. The latent representation of size (10) is reconstructed with a decoder consisting of fully connected layers of parameters $(10, 384)$ and $(384, 864)$ followed by three transpose convolutional layers of parameters $(48, 3, 2, 0, 0)$, $(24, 3, 2, 1, 1)$, and $(3, 3, 2, 1, 1)$. All the layers use ReLU activations, and we include batch normalization after the hidden convolutional and transpose convolutional layers.

C.2 Implementation of the compression priority function

The priority function used in our experiments (see Appendix B.1) is given by:

$$\mathcal{W}(\mathbf{x}|\bar{\theta}) = a_k \left[-\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2}{E(\mathbf{x})} \right]. \tag{8}$$

It has three main components: the reconstruction error, the energy term, and the activation function. The reconstruction error is the same as in the loss function. That is, when computing the loss of a single element (see Eq. (2) and Algorithm 1) we re-use the first term of the loss for the weighting function but drop the L1-regularization. However, in order to do so, we detach the loss term so as to prevent the gradient descent from being computed over the weighting terms instead of the loss itself.

Energy typically refers to a local change in intensity. The term differs between the image and audio experiments. For images, local changes can be revealed through convolutions with different types of filters, e.g. Laplacian, Gaussian, or Sobel filters. Here, we opt for a Laplacian filter of size 3×3 given by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \tag{9}$$

that detects sharp edges and is reminiscent of center-surround receptive fields of retinal ganglion cells. After convolving each input with this filter, we compute the absolute sum of the outcome as the total energy of the image:

$$E(\mathbf{x}) = \sum_{i=1}^{D} |\mathbf{x} * \mathbf{A}|_i \ . \tag{10}$$

For the audio experiments with ITD, the energy of the sound wave is measured directly by the total activation of the mel-spectrogram of the input $\mathcal{M}(\mathbf{x})$:

$$E(\mathbf{x}) = \sum_{i=1}^{D} |\mathcal{M}(\mathbf{x})|_i \ . \tag{11}$$

The total dimension of the inputs is given by $D = d \times 28 \times 28$, where $d = 1$ for the symmetry experiments, $d = 2$ for the disparity and ITD experiments, and $d = 3$ for the exposure onset experiments.

Finally, for the activation function, we use a k-winners-take-all, with $k = 50$ for a batch size 250 inputs. Doing so means the training is effectively performed on $1/5$ of the total training data. If the stochasticity prioritization ϵ is 0 (see B.3), then we assign a priority value of 1 to the k inputs with highest compression and 0 to the remaining inputs in the batch. If $\epsilon > 0$, we first assign values of 1 to the highest $k \times (1 - \epsilon)$, and then perform a uniform sample without replacement of $k \times \epsilon$ inputs within the rest that are also assigned a value of 1.

C.3 Baselines for comparison

Our PRL method is always compared with two different baselines. First, a so-called **full** model uses the entire batches during training. Hence, this is equivalent to using a N-winners-take-all activation, where N is the number of inputs in a batch. This baseline is trained on the same inputs as the PRL model, but also on additional 200 inputs per batch. Therefore, this baseline allows us to understand how important it is to save resources for the preferred inputs rather than trying to represent all possible inputs. The second baseline consists of a **uniform** sampling of 50 inputs within each batch, equivalent to performing a k-winners-take-all with $\epsilon = 1$ stochasticity. This baseline tackles the question of how important the prioritization of the weighted loss is, in comparison to randomly selected inputs. In this sense, comparisons between the PRL and uniform models may be more informative about the role of compression, since both are trained on the same amount of data. On the other hand, comparisons of both the PRL and uniform models with the full model can provide insights about the role of batch sizes in the learned representations.

C.4 Training

The training of the autoencoders is common to all datasets and models. The encoder and decoder are trained together using the loss from Eq. (2), or its corresponding priority-weighted version. The reconstruction error is computed as the mean squared error over the $D =$*channels*$\times$*height*$\times$*width* dimensions of the inputs. For sparsity, we compute the mean absolute value of all the activations of intermediate layers of the autoencoder after their corresponding ReLU activations (see model architectures above). In total, each autoencoder is trained for 30 epochs, where one epoch consists of presenting the entire training data once. We store the weights of the models every 5 epochs and later evaluate them on test data. All models use the Adam optimizer with a learning rate of $\gamma = 10^{-3}$ and a weight decay of $\lambda = 10^{-5}$. By default, we use batches of $N = 250$ inputs, $k = 50$ for the k-winners-take-all activations, and a sparsity ratio of $\eta = 0.01$.

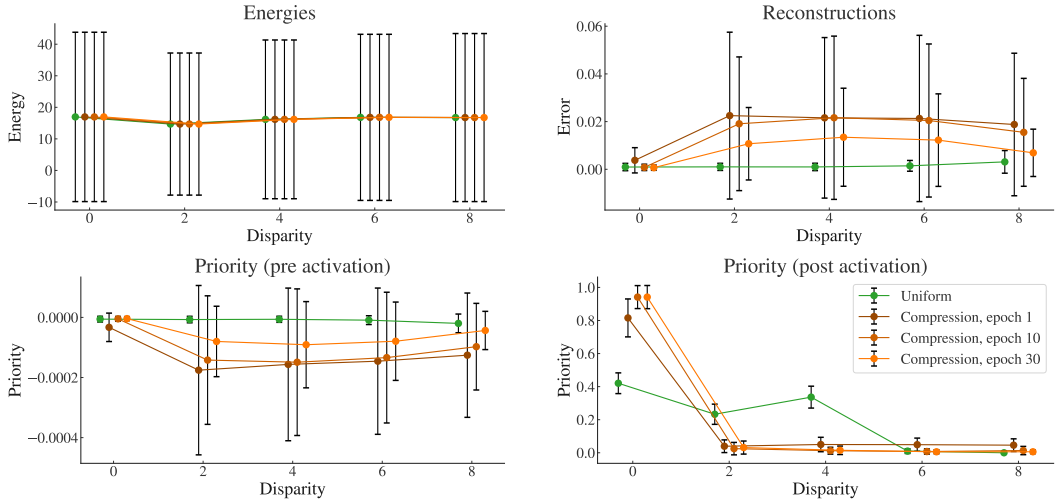
After training the autoencoders, we also train linear classifiers on the representations. To do so, we freeze the weights of each encoder and re-use the training dataset with its corresponding labels. However, unlike the training of the autoencoders, the classifiers are only trained and tested on "biased" data, i.e. the human-preferred conditions for each experiment: zero binocular disparity, both axes of symmetry, zero ITD, and red images for earliest exposure. The classifiers have a single layer mapping the latent space into a vector of size 10, corresponding to each digit of the MNIST images or audios, with an additional Softmax activation on the output. We use the Adam optimizer with a learning rate of $\gamma = 10^{-3}$. The training is done for 10 epochs, after which the performance of the classifier is evaluated on the test dataset.

All the experiments are repeated with three different seeds to randomize the initialization of the models and sampling of the datasets. All results showed are averages over the three sets of runs.

C.5 Additional results

Here we show additional results from our experiments. Unless specified, these experiments are performed with the binocular disparity dataset.

a) Dependence on binocular disparity



b) Dependence on input labels

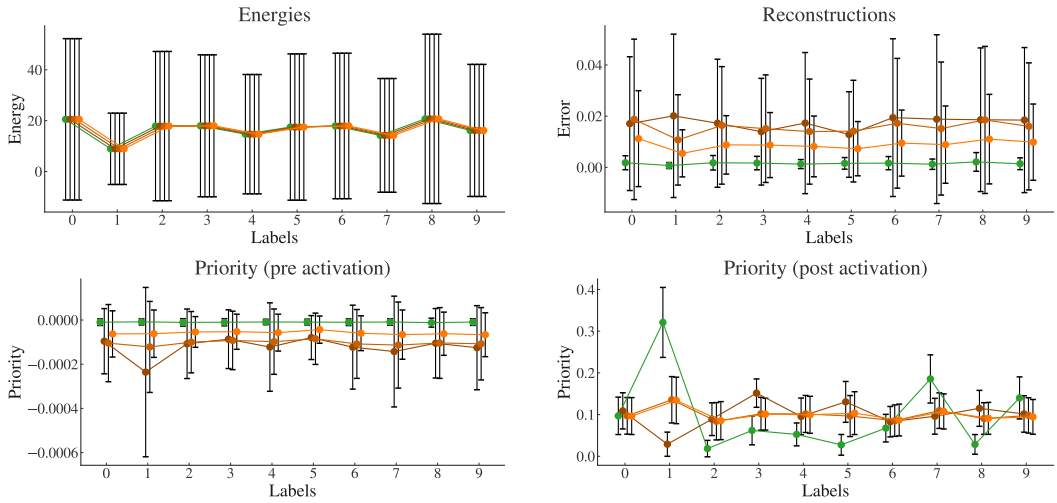


Figure 5: Analysis of the components of the priority function for a batch of test data as functions of the binocular disparity (top) and digit labels (bottom). We show the average energies, reconstruction errors, and priorities before and after the activation function for the PRL after 1, 10, and 30 epochs of training. For comparison, we also show the values that would be attained by a model trained with uniform sampling.

C.5.1 Compression priority function

The priority function is described in detail in Appendix C.2. It has three different components: the energy of the input, the reconstruction error based on the representation model, and a k-winners-take-all activation function. The contributions of each of these can be seen in Fig. 5. We use the entire test dataset and take batches with uniformly sampled binocular disparities. We plot the energies, reconstruction errors, and priorities before and after the activation function for each binocular disparity and label. We compare PRL models at different stages of training (1 epoch, 10 epochs, and 30 epochs) with a uniform model, even though the latter does not actually use any priority weighting during training.

The results are very informative about the interest in using PRL. First of all, as expected, we see that the energy of the inputs has only minor dependencies with disparity, but considerably larger dependencies with label identity. This is a known fact about the MNIST dataset: in particular, the

digits 1 and 7 are easy to encode compared to 6 or 8. All different models have the same energies because they are evaluated on the same data and with the same random manipulations. Looking at the reconstruction performance we find that, as expected, the error is lowest for the zero-disparity inputs with PRL, but there are only marginal differences for the uniform model. The result is that the priority function pre activation has a maximum for zero-disparity for all PRL models, even after a single epoch. Thus, the zero-disparity data is most used for training the PRL model, as evidenced in the priority function post activation. On the other hand, the uniform model would roughly divide its training data into three thirds, consisting of disparities of 0, 2, and 4 pixels.

As for the dependence on input labels, we find that there is some marginal variability in the priority of the PRL model, but the uniform model would select labels 1 and 7 more often than the rest. The fact that this effect is not seen in the PRL model is a testament to the priority for compression that is learned with PRL: our model is exploiting binocular redundancies rather than merely favoring inputs with low energy.

C.5.2 Learning curves

The learning curves for the main experiments are shown in Fig. 6. Each experiment has three plots. First, on the left, the training loss collected at each of the 30 epochs and defined by the weighted loss corresponding to each specific method. The middle plots show exclusively the reconstruction errors evaluated on the test datasets and with uniformly-sampled random manipulations. In this case, all three models are presented with the entire test data and no prioritization is possible, i.e. the reconstruction errors shown are averages for all the images or audios. The last plots show the reconstruction errors evaluated on the same data but where only the preferred manipulation is performed on all images.

Since PRL prioritizes inputs with high compression, and these typically have low reconstruction error, it is not surprising that PRL achieves lower training losses. This has an interesting consequence in the exposure onset experiment. At the sixth epoch, when the green images start to be shown to the model, its training loss increases, as does that of the uniform and full models. However, at the eleventh epoch, when blue images are introduced, the two baseline models have higher losses because they have to learn representations for this new type of input, whereas the PRL model can filter the blue images and focus exclusively on its preferred red and green inputs. Additionally, we also find that the full model has marginal advantages over the uniform sampling method across all experiments. For the former, the training loss is computed for all elements in a batch, whereas for the latter it is computed over a random fifth of the elements. Hence, we find that having access to the entire training datasets provides the full model with further training opportunities that the uniform model does not benefit from.

These marginal advantages of the full versus the uniform model also translate to the evaluations on the test datasets with random manipulations. On the other hand, the PRL models perform notoriously poorly when evaluated with all random manipulations. That is, even though the PRL model learns continuously throughout the experiment, it does not improve its performance on random evaluations, but rather exclusively on the preferred manipulations, as evidenced by the right column plots. This behavior is quite peculiar and highlights the main interest behind our work: while conventional passive approaches minimize a loss function over an entire dataset, PRL selectively filters which inputs are effectively interesting and worth representing accurately, and which ones should instead be ignored.

These results may appear somewhat underwhelming. The small advantage for the preferred manipulations comes at the cost of considerably higher reconstruction errors for all other manipulations. This is very much expected. PRL is not a method intended to improve representation learning everywhere, but rather to achieve representations that match those of humans and non-human animals. Therefore, the use of our model will not be of interest in many applications, mainly those where the goal is to minimize reconstruction errors and maximize performance in downstream tasks. Instead, PRL can be applied in settings aimed at recreating human representations and behaviors, such as those described in our experiments.

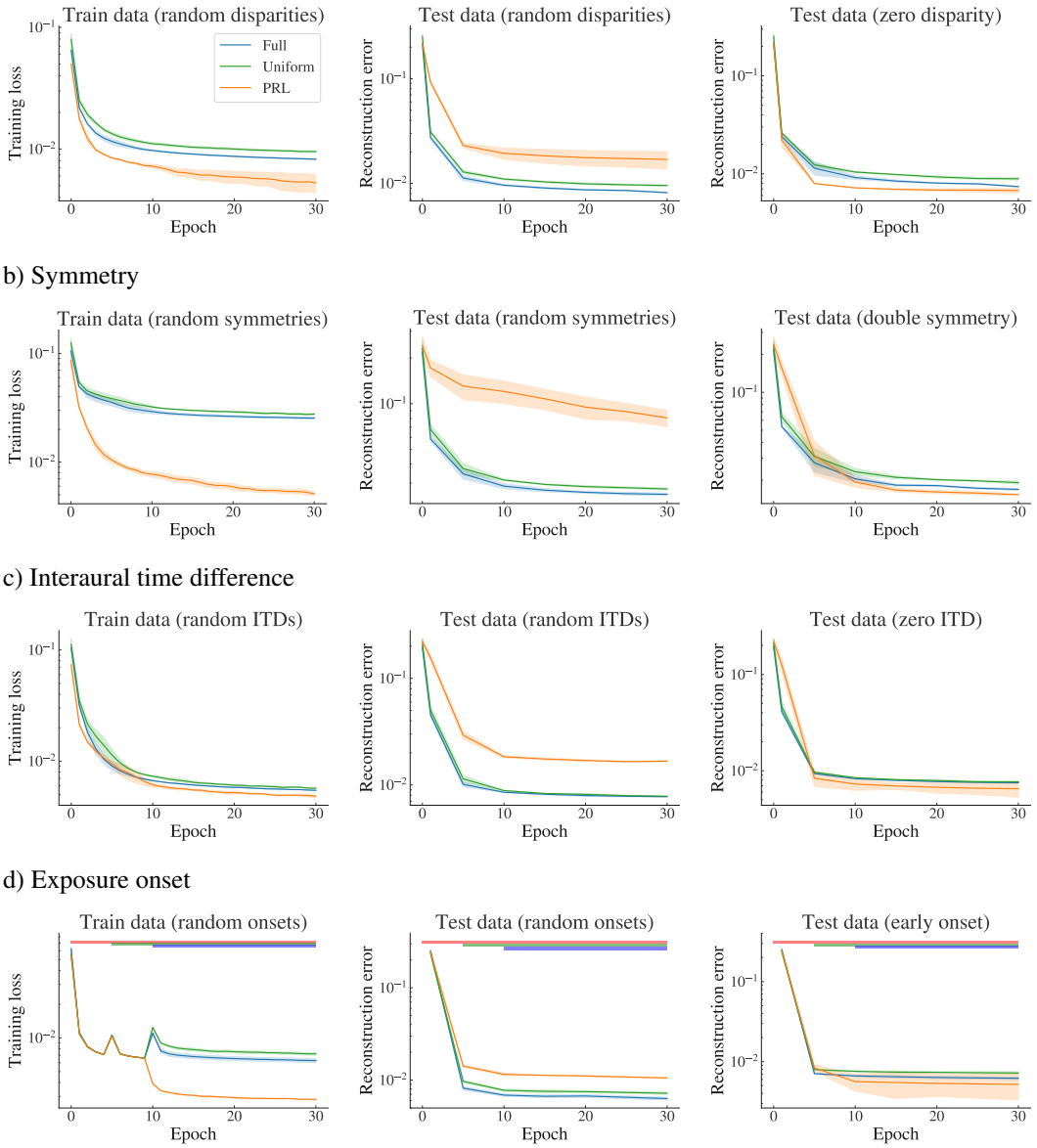
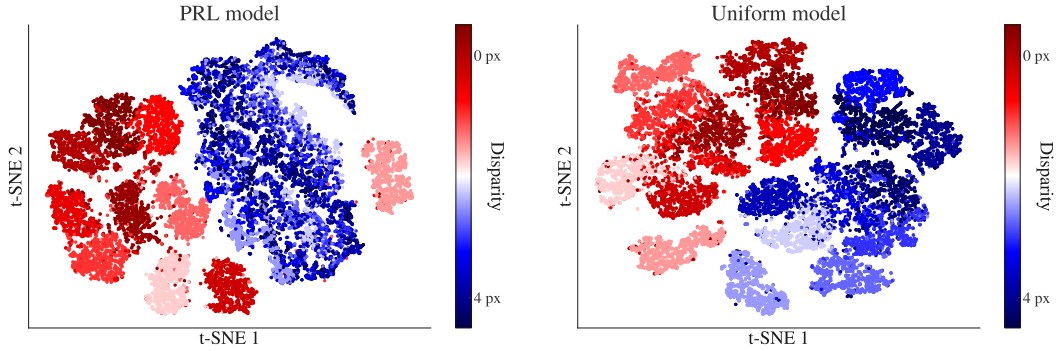


Figure 6: Learning curves. Left plots show the training loss computed on the training data with random manipulations. The plots on the center show the reconstruction errors for the test datasets with random manipulations, and the plots on the right show the reconstruction errors on the test datasets with the human-preferred manipulation. The horizontal color bars in the exposure onset plots indicate the colors of the inputs being shown to the model at each epoch.

C.5.3 Latent space

In Fig. 7 we present two different visualization of the latent spaces of the PRL and uniform models. First, we visualize the organization of the latent spaces using the t-SNE algorithm [71] for dimensionality reduction on the entire test dataset at two binocular disparities: 0 and 4 pixels. The plots reveal that both models achieve clustered representations that can differentiate between zero disparity and higher disparities. Furthermore, we see that the uniform model successfully learns clustered representations for the labels (i.e. the digits) of the dataset at both disparities. On the other hand, PRL successfully clusters the representations at zero disparity but fails at the higher disparity. Interestingly,

a) Dimensionality reduction visualizations



b) Images generated from latent space

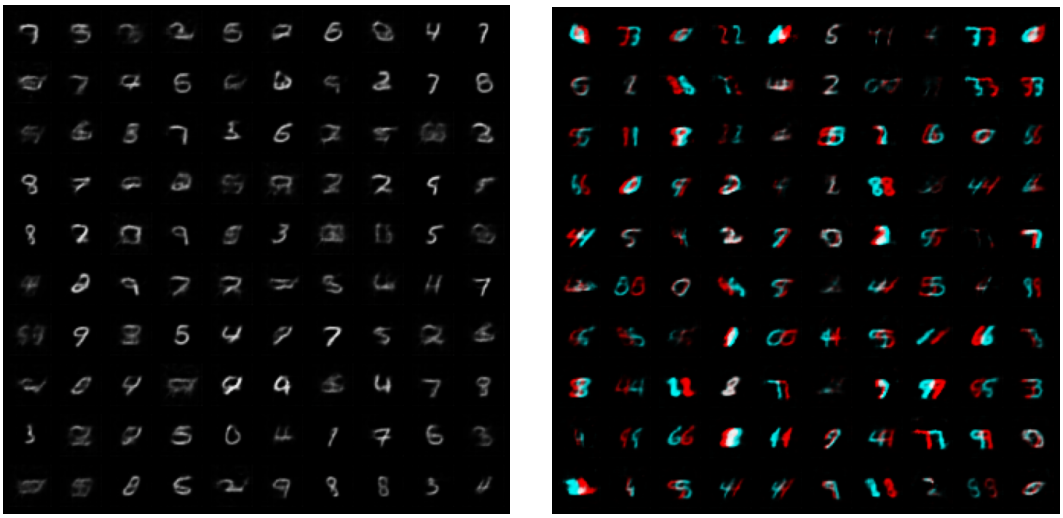


Figure 7: Visualizations of the latent space for the PRL (left) and uniform (right) models trained on the binocular disparity dataset. On the top we show the organization of the latent representations using the t-SNE algorithm [71]. Red dots indicate images with zero disparity and blue dots images with 4 pixels of disparity. Color intensity corresponds to image label. On the bottom, we generate images with random latent vectors and the decoders of the two models.

the clusters for the zero-disparity inputs of the PRL model appear to be more defined than those of the uniform model, which explains the higher accuracy achieved by the linear classifiers (see Fig. 3).

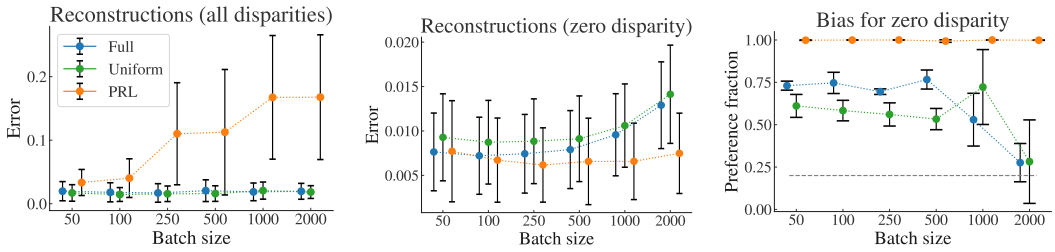
Fig. 7.b presents some synthetically generated inputs by feeding random latent vectors into the decoders of the two models. A striking difference can be observed: while the PRL model exclusively generates binocular images with zero disparity, the uniform model mostly outputs double-images with varying degrees of disparity. Once again, these plots highlight how the PRL model is fine-tuned for its preferred inputs and does not – rather cannot – accurately represent inputs with high disparity.

C.5.4 Hyperparameters analysis

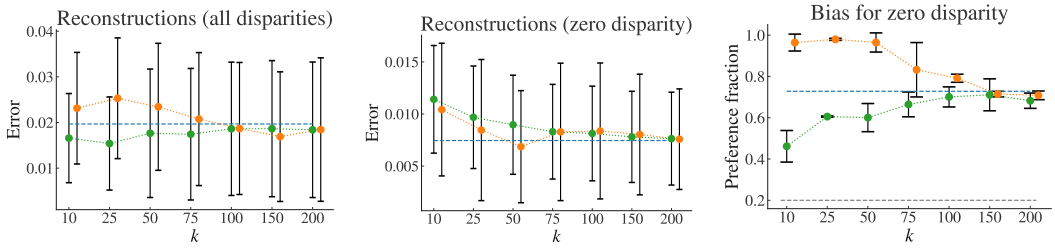
Lastly, we analyze the PRL, uniform, and full models when changing different training parameters. All experiments are carried out with the binocular disparity dataset. In Fig. 8 we report the reconstruction errors averaged for all disparities, as well as the reconstruction errors restricted to the zero-disparity inputs and the bias for these. When not possible, e.g. the evaluation of the k-winners-take-all with the full model, we plot the baseline results for comparison.

Batch size For the different batch sizes, we always use $1/5$ of the batch in the k-winners-take-all activation function, both for the PRL and uniform models. Overall, we find that the PRL models

a) Batch size



b) k-winners-take-all



c) Latent dimension

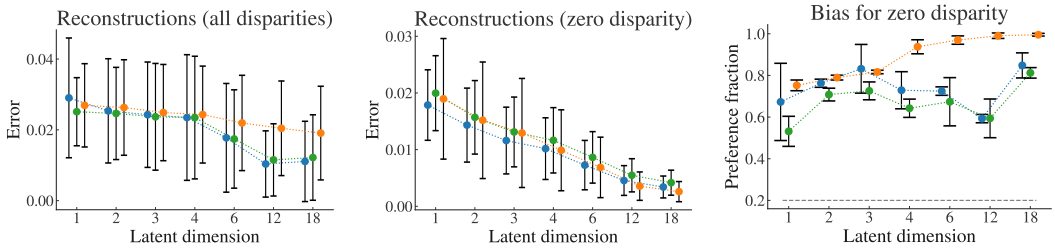


Figure 8: Analysis of model hyperparameters. Left plots show the average reconstruction errors for the test dataset with all binocular disparities, and the center plots only with zero-disparity. Right plots show the bias for zero disparity. Colored dashed lines without markers are baseline results for comparison. Gray dashed lines indicate change levels.

performs considerably worse with large batches than with smaller batches when evaluated on all disparities, whereas the opposite is true for the full and uniform models. Larger batch sizes result in a reduced performance on the zero-disparity inputs for the uniform and full models, whereas the outcome of the PRL approach is mostly unchanged. Even at a batch size of 2000 inputs, since PRL is effectively only training on 400 of those inputs, it can accurately compute its loss function on the preferred inputs, at the expense of the representations of all other inputs. Interestingly, the bias for zero disparity of the PRL is not affected by the batch size, but it disappears almost entirely for the other two models with too large batches.

Activation k-winners-take-all The k-winners-take-all has little effect on the reconstruction error for the uniform model, but it qualitatively changes the behavior of the PRL model. Up to $k = 100$, PRL learns poor general representations and hence achieves a bias for zero disparity beyond the threshold of the full model. However, for $k > 100$, the difference in bias for zero disparity disappears. That is, if the training is done on large portions of the batches, then all possible disparities contribute to the loss function. For our main experiments we use $k = 50$, which results in the largest bias advantage. This value, corresponding to $1/5$ of each batch, is larger than the the amount of zero-disparity inputs in a batch.

Latent dimension The dimension of the latent space should be the main bottleneck for the representations learned with an autoencoder. In Fig. 8.c we show that increasing the latent dimension does

in fact reduce the reconstruction errors, although the differences are quantitative and not qualitative. Likewise, and contrary to our initial expectations, the PRL model outperforms the full and uniform models only for latent codes of length 6 or larger, and even then the differences are moderate. The model architecture is achieving satisfactory performances even with very small latent spaces. This is likely due to the use of three convolutional layers, admittedly an excessive computational power for a dataset as simple as MNIST. In future research we will explore whether these results are maintained when using a simpler autoencoder architecture, e.g. a sparse autoencoder with a single hidden layer.

Stochastic prioritization We introduce stochastic prioritization drawing inspiration from PER [5]. The results of this analysis are shown in Fig. 4. In our experiments, we find that even a small stochastic prioritization $\epsilon = 0.04$ can result in comparable reconstruction errors to the full and uniform models for all binocular disparities while retaining the lower reconstruction error with zero disparity and a bias close to 1. Of course, as the stochastic prioritization increases further, the results get closer to the uniform model. The optimal stochasticity may also depend on the experiment and the other hyperparameters.