EVALUATING REPRESENTATIONS BY THE COMPLEXITY OF LEARNING LOW-LOSS PREDICTORS

William F. Whitney[*] **Min Jae Song** **David Brandfonbrener** **Jaan Altosaar**

Kyunghyun Cho

ABSTRACT

We consider the problem of evaluating representations of data for use in solving a downstream task. We propose to measure the quality of a representation by the complexity of learning a predictor on top of the representation that achieves low loss on a task of interest. To this end, we introduce two measures: surplus description length (SDL) and ε sample complexity (εSC). To compare our methods to prior work, we also present a framework based on plotting the validation loss versus evaluation dataset size (the “loss-data” curve). Existing measures, such as mutual information and minimum description length, correspond to slices and integrals along the data axis of the loss-data curve, while ours correspond to slices and integrals along the loss axis. This analysis shows that prior methods measure properties of an evaluation dataset of a specified size, whereas our methods measure properties of a predictor with a specified loss. We conclude with experiments on real data to compare the behavior of these methods over datasets of varying size.

1 INTRODUCTION

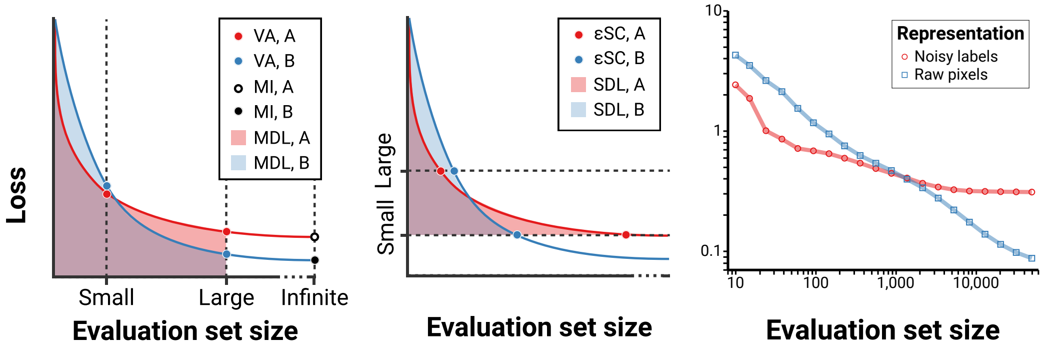
One of the first steps in building a machine learning system is selecting a representation of data. Huge strides in unsupervised learning (Hénaff et al., 2020; Chen et al., 2020; He et al., 2019; van den Oord et al., 2018; Bachman et al., 2019; Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2019; Brown et al., 2020) have led to common wisdom now recommending that the design of most systems start from a pretrained representation rather than learning representations from scratch in an end-to-end manner. With this boom in representation learning techniques, practitioners and representation researchers alike have the question: Which representation is best for my task?

We take the position that the best representation is the one which allows for learning a predictor to solve the task using the least label information. This quality is measured on an *evaluation dataset* which may be distinct from the set of downstream applications. This position is motivated by practical concerns; the more labels that are needed to solve a new task, the more expensive to use and the less widely applicable a representation will be.

Recent works (Voita & Titov, 2020) have proposed to use an estimate of the minimum description length (MDL) of a dataset's labels given a representation as a measure of representation quality. However, this code length perspective conflates the information required for learning a model with the irreducible uncertainty in the data. We show that this leads MDL-based measures to choose different representations when given evaluation datasets of different sizes, potentially leading to a premature decision about which representation to use.

We propose a new measure of representation quality called *surplus description length* (SDL), which modifies the MDL to measure the complexity of learning an ε-loss predictor rather than computing the complexity of the labels in the evaluation dataset. SDL, along with a second measure called *ε-sample complexity* (εSC), provide tools for researchers and practitioners to evaluate the extent to which a learned representation can improve data efficiency. Furthermore, they formalize existing research challenges for learning representations which allow state of the art performance while using as few labels as possible (e.g. Hénaff et al. (2020)). To facilitate our analysis, we also propose a

[*] wwhitney@cs.nyu.edu



(a) Existing measures (b) Proposed measures (c) Illustrative experiment

Figure 1: Each measure for evaluating representation quality is a simple function of the "loss-data" curve shown here, which plots validation loss of a probe against evaluation dataset size. **Left:** Validation accuracy (VA), mutual information (MI), and minimum description length (MDL) measure properties of a given evaluation dataset, with VA measuring the loss at a finite amount of evaluation data, MI measuring it at infinity, and MDL integrating it from zero to n. This dependence on evaluation dataset size can lead to misleading conclusions as the amount of available data changes. **Middle:** Our proposed methods instead measure the complexity of learning a predictor with a particular loss tolerance. ε sample complexity (εSC) measures the number of samples required to reach that loss tolerance, while surplus description length (SDL) integrates the surplus loss incurred above that tolerance. Neither depends on the evaluation dataset size. **Right:** A simple example task which illustrates the issue. One representation, which consists of noisy labels, allows quick learning, while the other supports low loss in the limit of data. Evaluating either representation at a particular evaluation dataset size risks drawing the wrong conclusion.

framework called the *loss-data framework*, illustrated in Figure 1, that plots the validation loss against the evaluation dataset size (Talmor et al., 2019; Yogatama et al., 2019; Voita & Titov, 2020).

2 The loss-data framework for representation evaluation

Notation. We use bold letters to denote random variables. A supervised learning problem is defined by a joint distribution $\mathcal{D}$ over observations and labels $(\mathbf{X}, \mathbf{Y})$ in the sample space $\mathcal{X} \times \mathcal{Y}$ with density denoted by p. Let the random variable $\mathbf{D}^n$ be a sample of n i.i.d. $(\mathbf{X}, \mathbf{Y})$ pairs, realized by $D^n = (X^n, Y^n) = \{(x_i, y_i)\}_{i=1}^n$. This is the evaluation dataset. Let $\mathcal{R}$ denote a representation space and $\phi : \mathcal{X} \to \mathcal{R}$ a representation function. The methods we consider all use parametric probes, which are neural networks $\hat{p}_\theta : \mathcal{R} \to P(\mathcal{Y})$ parameterized by $\theta \in \mathbb{R}^d$ that are trained on D^n to estimate the conditional distribution $p(y \mid x)$. We often abstract away the details of learning the probe by simply referring to an algorithm $\mathcal{A}$ which returns a predictor: $\hat{p} = \mathcal{A}(\phi(D^n))$. Abusing notation, we denote the composition of $\mathcal{A}$ with ϕ by $\mathcal{A}_\phi$. Define the population loss and the expected population loss for $\hat{p} = \mathcal{A}_\phi(D^n)$, respectively as

$$L(\mathcal{A}_\phi, D^n) = \mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} - \log \hat{p}(\mathbf{Y} \mid \mathbf{X}) \qquad\qquad L(\mathcal{A}_\phi, n) = \mathop{\mathbb{E}}_{\mathbf{D}^n} L(\mathcal{A}_\phi, \mathbf{D}^n). \tag{1}$$

The representation evaluation problem. The representation evaluation problem asks us to define a real-valued measurement of the quality of a representation ϕ for solving solving the task defined by $(\mathbf{X}, \mathbf{Y})$. Explicitly, each method defines a real-valued function $m(\phi, \mathcal{D}, \mathcal{A}, \Psi)$ of a representation ϕ, data distribution $\mathcal{D}$, probing algorithm $\mathcal{A}$, and some method-specific set of hyperparameters Ψ. By convention, smaller values of the measure m correspond to better representations. Defining such a measurement allows us to compare different representations.

2.1 Defining the loss-data framework

The loss-data framework is a lens through which we contrast different measures of representation quality. The key idea, demonstrated in Figure 1, is to plot the loss $L(\mathcal{A}_\phi, n)$ against the evaluation dataset size n. Explicitly, at each n, we train a probing algorithm $\mathcal{A}$ using a representation ϕ to produce a predictor $\hat{p}$, and then plot the loss of $\hat{p}$ against n. We can represent prior measures as points on the curve at fixed x (validation accuracy with a small dataset, VA) or integrals of the curve along the x-axis (MDL). Our measures correspond to evaluating points or integrals on the y-axis (SDL and εSC).

2.2 Existing methods in the loss-data framework

Validation accuracy and mutual information Two popular methods for evaluating representation quality are the validation accuracy of a model trained on limited data and the mutual information between the labels and a representation. As we describe in Appendix D, these measures are intimately linked. In the loss-data framework, we formally define the validation accuracy and mutual information measures respectively as

$$m_{\mathrm{VA}}(\phi, \mathcal{D}, \mathcal{A}, n) = L(\mathcal{A}_\phi, n) \qquad\qquad m_{\mathrm{MI}}(\phi, \mathcal{D}, \mathcal{A}) = \lim_{n\to\infty} L(\mathcal{A}_\phi, n). \tag{2}$$

Minimum description length. Recent studies (Yogatama et al., 2019; Voita & Titov, 2020) propose using the Minimum Description Length (MDL) principle (Rissanen, 1978; Grünwald, 2004) to evaluate representations. These works use an online or prequential code (Blier & Ollivier, 2018) to encode the labels given the representations. The codelength ℓ of Y^n given $\phi(X^n)$ is then defined as

$$\ell(Y^n \mid \phi(X^n)) = -\sum_{i=1}^{n} \log \hat{p}_i(y_i \mid \phi(x_i)), \tag{3}$$

where $\hat{p}_i$ is the output of running a pre-specified algorithm $\mathcal{A}$ on the evaluation dataset up to element i: $\hat{p}_i = \mathcal{A}_\phi(X^n_{1:i}, Y^n_{1:i})$. This measure can exhibit large variance on small evaluation datasets, especially since it is sensitive to the (random) order in which the examples are presented. We remove this variance by taking an expectation over the sampled evaluation datasets for each i and define a population variant of the MDL measure (Voita & Titov, 2020) as

$$m_{\mathrm{MDL}}(\phi, \mathcal{D}, \mathcal{A}, n) = \mathbb{E}\left[\ell(\mathbf{Y}^n \mid \phi(\mathbf{X}^n))\right] = \sum_{i=1}^{n} L(\mathcal{A}, i). \tag{4}$$

Thus, m_{MDL} measures the area under the loss-data curve on the interval $x \in [0, n]$.

3 Surplus description length (SDL)

The prequential code for estimating MDL computes the description length of the labels given observations in an evaluation dataset by iteratively creating tighter approximations $\hat{p}_1 \dots \hat{p}_n$ and integrating the area under the curve. Examining Equation (4), we see that $m_{\mathrm{MDL}}(\phi, \mathcal{D}, \mathcal{A}, n) \geq \sum_{i=1}^{n} H(\mathbf{Y} \mid \phi(\mathbf{X}))$. If $H(\mathbf{Y} \mid \phi(\mathbf{X})) > 0$, MDL grows without bound as the size of the evaluation dataset n increases. This poses a problem for representation evaluation, since simply evaluating the same representations on the same task, but using an evaluation dataset of a different size, may change the verdict about which representation is better. We call this issue, which VA shares, *sensitivity to evaluation dataset size*.

To derive an improved measure we look to information theory. Imagine trying to efficiently encode a large number of samples of a random variable $\mathbf{e}$ which takes values in $\{1 \dots K\}$ with probability $p(\mathbf{e})$. An optimal code for these events has expected length[1] $\mathbb{E}[\ell(\mathbf{e})] = \mathbb{E}_{\mathbf{e}}[-\log p(\mathbf{e})] = H(\mathbf{e})$. If this data is instead encoded using a probability distribution $\hat{p}$, the expected length becomes $H(\mathbf{e}) + D_{\mathrm{KL}}\big(p \mid\mid \hat{p}\big)$. We call $D_{\mathrm{KL}}\big(p \mid\mid \hat{p}\big)$ the *surplus description length* (SDL) from encoding according to $\hat{p}$ instead of p:

$$D_{\mathrm{KL}}\big(p \mid\mid \hat{p}\big) = \mathop{\mathbb{E}}_{\mathbf{e}\sim p}\left[\log p(\mathbf{e}) - \log \hat{p}(\mathbf{e})\right]. \tag{5}$$

We propose to measure the complexity of a learned predictor $p(\mathbf{Y} \mid \phi(\mathbf{X}))$ by computing the surplus description length of encoding an infinite stream of data according to the online code instead of the true conditional distribution; when $\mathcal{A}$ converges to the true predictor, this measure is bounded.

[1] in nats

Definition 1 (Surplus description length of online codes)**.** *Given random variables* $\mathbf{X}, \mathbf{Y} \sim \mathcal{D}$*, a representation function* ϕ*, and a learning algorithm* $\mathcal{A}$*, define*

$$m_{\text{SDL}}(\phi, \mathcal{D}, \mathcal{A}) = \sum_{i=1}^{\infty} \Big[L(\mathcal{A}_\phi, i) - H(\mathbf{Y} \mid \mathbf{X}) \Big]. \tag{6}$$

We generalize this definition to measure the complexity of learning an approximating conditional distribution with loss ε. This corresponds to the additional description length incurred by encoding data with the learning algorithm $\mathcal{A}$ rather than using a fixed predictor with loss ε.

Definition 2 (Surplus description length of online codes with a specified baseline)**.** *Take random variables* $\mathbf{X}, \mathbf{Y} \sim \mathcal{D}$*, a representation function* ϕ*, a learning algorithm* $\mathcal{A}$*, and a loss tolerance* $\varepsilon \geq H(\mathbf{Y} \mid \mathbf{X})$*. Let* $[c]_+$ *denote* $\max(0, c)$ *and then we define*

$$m_{\text{SDL}}(\phi, \mathcal{D}, \mathcal{A}, \varepsilon) = \sum_{i=1}^{\infty} \Big[L(\mathcal{A}_\phi, i) - \varepsilon \Big]_+. \tag{7}$$

One interpretation of this measure is that it gives the cost (in terms of information) for re-creating an ε-loss predictor when using the representation ϕ. In our framework, the surplus description length corresponds to computing the area between the loss-data curve and a baseline set by $y = \varepsilon$. Whereas MDL measures the complexity of a sample of n points, SDL measures the complexity of a function which solves the task to ε tolerance. For guidance about setting ε and computing SDL, please see Appendix B.

3.1 ε SAMPLE COMPLEXITY (εSC)

We also introduce a second measure called ε sample complexity (εSC), which consists of the number of samples needed to reach an expected population loss of ε. We discuss this measure in Appendix A.

4 EXPERIMENTS

We empirically show the behavior of VA, MDL, SDL, and εSC with experiments on MNIST. The results, shown in Figure 2, demonstrate that the issue of sensitivity to evaluation dataset size in fact occurs in practice: VA and MDL choose different representations when given evaluation sets of different sizes. Because these measures are a function of the evaluation dataset size, making a decision about which representation to use with a small evaluation dataset would be premature. By contrast, SDL and εSC are functions only of the data *distribution*, not a finite sample. Once they measure the complexity of learning an ε-loss function, that measure is invariant to the size of the evaluation dataset.

We also performed experiments on part of speech classification with representations from ELMo (Peters et al., 2018). The results on this large-scale task, which are largely the same as those shown here, are in Appendix C. Details of the experiments, including representation training, probe architectures, and hyperparameters, are available in Appendix H.

n	Representation	CIFAR	Pixels	VAE
60	VA	0.88	1.54	**0.70**
	MDL	122.75	147.34	**93.8**
	SDL, ε=0.1	> 116.75	> 141.34	> 87.8
	εSC, ε=0.1	> 60.0	> 60.0	> 60.0
31936	VA	**0.05**	0.10	0.13
	MDL	**2165.1**	5001.57	4898.37
	SDL, ε=0.1	**260.6**	1837.08	> 1704.77
	εSC, ε=0.1	**3395**	31936	> 31936.0

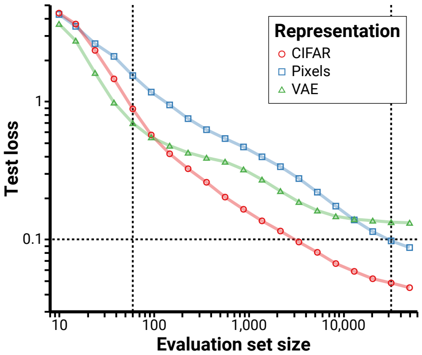


Figure 2: Estimated measures of representation quality on MNIST. At small evaluation dataset sizes, VA and MDL state that the VAE representation is the best, even though every representation yields poor predictions with that amount of data. Since SDL and εSC have a target for prediction quality, they are able to report when the evaluation dataset is insufficient to achieve the desired performance.

References

Bachman, P., Hjelm, R. D., and Buchwalter, W. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, 2019.

Blier, L. and Ollivier, Y. The description length of deep learning models. In *Advances in Neural Information Processing Systems*, 2018.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCand lish, S., Radford, A., Sutskever, I., and Amodei, D. Language Models are Few-Shot Learners. *arXiv preprint arXiv:2005.14165*, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. E. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, pp. 4171–4186, 2019.

Dubois, Y., Kiela, D., Schwab, D. J., and Vedantam, R. Learning optimal representations with the decodable information bottleneck. *ArXiv*, abs/2009.12789, 2020.

Grünwald, P. A tutorial introduction to the minimum description length principle. *arXiv preprint math:0406077*, 2004.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Hénaff, O. J., Srinivas, A., Fauw, J., Razavi, A., Doersch, C., Eslami, S., and Oord, A. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020.

Hewitt, J. and Liang, P. Designing and interpreting probes with control tasks. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, pp. 2733–2743, 2019.

Hill, F., Cho, K., and Korhonen, A. Learning distributed representations of sentences from unlabelled data. In *HLT-NAACL*, 2016.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *ArXiv*, abs/2001.08361, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. *International Conference on Learning Representations*, 2014.

Kiros, R., Zhu, Y., Salakhutdinov, R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. Skip-thought vectors. In *NIPS*, 2015.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

McAllester, D. and Stratos, K. Formal limitations on the measurement of mutual information. *International Conference on Artificial Intelligence and Statistics*, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. In *North American Chapter of the Association for Computational Linguistics*, pp. 2227–2237, 2018.

Pimentel, T., Valvoda, J., Maudslay, R. H., Zmigrod, R., Williams, A., and Cotterell, R. Information-theoretic probing for linguistic structure. *arXiv preprint arXiv:2004.03061*, 2020.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019.

Resnick, C., Zhan, Z., and Bruna, J. Probing the state of the art: A critical look at visual representation evaluation. *arXiv preprint arXiv:1912.00215*, 2019.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Rissanen, J. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

Talmor, A., Elazar, Y., Goldberg, Y., and Berant, J. oLMpics–on what language model pre-training captures. *arXiv preprint arXiv:1912.13283*, 2019.

van den Oord, A., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Voita, E. and Titov, I. Information-theoretic probing with minimum description length. *arXiv preprint arXiv:2003.12298*, 2020.

Xu, Y., Zhao, S., Song, J., Stewart, R., and Ermon, S. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020.

Yogatama, D., d'Autume, C. d. M., Connor, J., Kocisky, T., Chrzanowski, M., Kong, L., Lazaridou, A., Ling, W., Yu, L., Dyer, C., et al. Learning and evaluating general linguistic intelligence. *arXiv preprint arXiv:1901.11373*, 2019.

Zhang, K. and Bowman, S. Language modeling teaches you more than translation does: Lessons learned through auxiliary syntactic task analysis. In *EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 359–361, 2018.

APPENDIX A ε SAMPLE COMPLEXITY (εSC)

In addition to surplus description length we introduce a second, conceptually simpler measure of representation quality: ε sample complexity.

Definition 3 (Sample complexity of an ε-loss predictor)**.** *Given random variables* $\mathbf{X}, \mathbf{Y} \sim \mathcal{D}$*, a representation function* ϕ*, a learning algorithm* $\mathcal{A}$*, and a loss tolerance* $\varepsilon \geq H(\mathbf{Y} \mid \phi(\mathbf{X}))$*, define*

$$m_{\varepsilon\mathrm{SC}}(\phi, \mathcal{D}, \mathcal{A}, \varepsilon) = \min\Big\{n \in \mathbb{N} : L(\mathcal{A}_\phi, n) \leq \varepsilon\Big\}. \tag{8}$$

The ε sample complexity measures the complexity of learning an ε-loss predictor by the number of samples it takes to find it. This measure allows the comparison of two representations by first picking a target function to learn (via a setting of ε), then measuring which representation enables learning that function with less data.

In our framework, sample complexity corresponds to taking a horizontal slice of the loss-data curve at $y = \varepsilon$, analogous to VA's slice at $y = n$. VA makes a statement about the data (by setting n) and reports the accuracy of some function given that data. In contrast, ε sample complexity specifies the desired function and determines its complexity by how many samples are needed to learn it.

Estimating the εSC. Given an assumption that algorithms are monotonically improving such that $L(\mathcal{A}, n+1) \leq L(\mathcal{A}, n)$, εSC can be estimated efficiently. With n finite samples in the evaluation dataset, an algorithm may estimate εSC by splitting the data into k uniform-sized bins and estimating $L(\mathcal{A}, {}^{ik}\!/_{n})$ for $i \in \{1 \ldots k\}$. By recursively performing this search on the interval which contains the transition from $L > \varepsilon$ to $L < \varepsilon$, we can rapidly reach a precise estimate or report that $m_{\varepsilon\mathrm{SC}}(\phi, \mathcal{D}, \mathcal{A}, \varepsilon) > n$. A more detailed examination of the algorithmic considerations of estimating εSC is in Appendix G, and an implementation is available in the supplement.

Using objectives other than negative log-likelihood. Our exposition of εSC uses negative log-likelihood for consistency with other methods, such as MDL, which require it. However, it is straightforward to extend εSC to work with whatever objective function is desired under the assumption that said objective is monotone with increasing data when using algorithm $\mathcal{A}$. A natural choice in many cases would be prediction accuracy, where a practitioner might target e.g. a 95% accurate predictor.

APPENDIX B SETTING ε AND ESTIMATING SDL

B.1 SETTING ε

A value for the threshold ε corresponds to the set of ε-loss predictors that a representation should make easy to learn. Choices of $\varepsilon \geq H(\mathbf{Y} \mid \mathbf{X})$ represent attainable functions, while selecting $\varepsilon < H(\mathbf{Y} \mid \mathbf{X})$ leads to unbounded SDL and εSC for any choice of the algorithm $\mathcal{A}$.

For evaluating representation learning methods in the research community, we recommend using SDL and establishing benchmarks which specify (1) a downstream task, in the form of an evaluation dataset; (2) a criterion for success, in the form of a setting of ε; (3) a standard probing algorithm $\mathcal{A}$. The setting of ε can be done by training a large model on the raw representation of the full evaluation dataset and using its validation loss as ε when evaluating other representations. This guarantees that $\varepsilon \geq H(\mathbf{Y} \mid \mathbf{X})$ and the task is feasible with any representation at least as good as the raw data. In turn, this ensures that SDL is bounded.

In practical applications, ε should be a part of the design specification for a system. As an example, a practitioner might know that an object detection system with 80% per-frame accuracy is sufficient and labels are expensive. For this task, the best representation would be one which enables the most sample efficient learning of a predictor with error $\varepsilon = 0.2$ using a 0 – 1 loss.

B.2 ESTIMATING SDL

Naively computing SDL would require unbounded data and the estimation of $L(\mathcal{A}_\phi, i)$ for every i. However, any reasonable learning algorithm obtains a better-generalizing predictor when given more i.i.d. data from the target distribution (Kaplan et al., 2020). If we assume that algorithms are monotonically improving so that $L(\mathcal{A}, i+1) \leq L(\mathcal{A}, i)$, SDL only depends on i up to the first point where $L(\mathcal{A}, n) \leq \varepsilon$. Approximating this integral can be done efficiently by taking a log-uniform partition of the evaluation dataset size and computing the Riemann sum as in Voita & Titov (2020). Note that evaluating a representation only requires training probes, not the large representation functions themselves, and thus has modest computational requirements. Crucially, if the tolerance ε is set unrealizeably low or the amount of available data is insufficient, an implementation is able to report that the given complexity estimate is only a lower bound. In Appendix F we provide a detailed algorithm for estimating SDL and a theorem proving its data requirements, and the supplement includes an implementation.

APPENDIX C EXPERIMENT ON PART OF SPEECH CLASSIFICATION

For a second experiment, shown in Figure 3, we compare the representations given by different layers of a pretrained ELMo model (Peters et al., 2018). We use the part-of-speech task introduced by Hewitt & Liang (2019) and implemented by Voita & Titov (2020) with the same probe architecture and other hyperparameters as those works. This leads to a large-scale representation evaluation task, with 4096-dimensional representation vectors and an output space of size 48^k for a sentence of k words.

The results demonstrate that SDL and εSC can scale to tasks of a practically relevant size. This experiment is of a similar size to the widespread use of BERT (Devlin et al., 2019) or SimCLR (Chen et al., 2020), and evaluating our measures to high precision took about an hour on one GPU.

	ELMo layer	0	1	2
n				
461	VA	0.75	**0.74**	0.87
	MDL	**884.54**	1009.26	1017.72
	SDL, ε=0.1	> 478.67	> 528.51	> 561.7
	εSC, ε=0.1	> 461	> 461	> 461
474838	VA	0.17	**0.08**	0.09
	MDL	92403.41	**52648.50**	65468.54
	SDL, ε=0.1	> 40882.72	**2765.11**	7069.56
	εSC, ε=0.1	> 474838	**237967**	474838

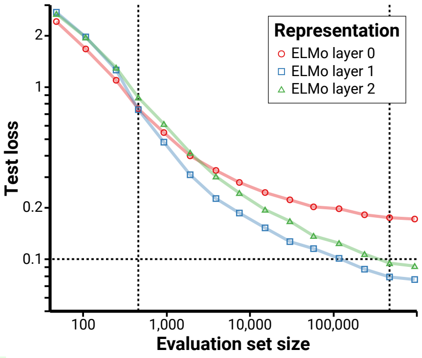


Figure 3: Estimated measures of representation quality on the part of speech classification task. With small evaluation datasets, MDL finds that the lowest ELMo layer gives the best results, but when the evaluation dataset grows the outcome changes.

APPENDIX D DERIVING VA AND MI MEASURES

Nonlinear probes with limited data. A simple strategy for evaluating representations is to choose a probe architecture and train it on a limited amount of data from the task and representation of interest (Hénaff et al., 2020; Zhang & Bowman, 2018); we call this the validation accuracy (VA) measure. This method can be interpreted in our framework by replacing the validation accuracy with the validation loss and taking an expectation over draws of evaluation datasets of size n. On the loss-data curve, this measure corresponds to evaluation at $x = n$:

$$m_{\text{VA}}(\phi, \mathcal{D}, \mathcal{A}, n) = L(\mathcal{A}_\phi, n) \tag{9}$$

Mutual information Mutual information (MI) between a representation $\phi(\mathbf{X})$ and targets $\mathbf{Y}$ is an often-proposed metric for learning and evaluating representations (Pimentel et al., 2020; Bachman et al., 2019). In terms of entropy, mutual information is equivalent to the information gain about $\mathbf{Y}$ from knowing $\phi(\mathbf{X})$:

$$I(\phi(\mathbf{X}); \mathbf{Y}) = H(\mathbf{Y}) - H(\mathbf{Y} \mid \phi(\mathbf{X})). \tag{10}$$

In general mutual information is intractable to estimate for high-dimensional or continuous-valued variables (McAllester & Stratos, 2020), and a common approach is to use a very expressive model for $\hat{p}$ and maximize a variational lower bound:

$$I(\phi(\mathbf{X});\mathbf{Y}) \geq H(\mathbf{Y}) + \mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})} \log \hat{p}(\mathbf{Y} \mid \phi(\mathbf{X})). \tag{11}$$

Since $H(\mathbf{Y})$ is not a function of the parameters, maximizing the lower bound is equivalent to minimizing the negative log-likelihood. Moreover, if we assume that $\hat{p}$ is expressive enough to represent p and take $n \to \infty$, this inequality becomes tight. As such, MI estimation can be seen a special case of nonlinear probes as described above, where instead of choosing some particular setting of n we push it to infinity. We formally define the mutual information measure of a representation as

$$m_{\mathrm{MI}}(\phi, \mathcal{D}, \mathcal{A}) = \lim_{n\to\infty} L(\mathcal{A}_\phi, n). \tag{12}$$

A decrease in this measure reflects an increase in the mutual information. On the loss-data curve, this corresponds to evaluation at $x = \infty$.

Appendix E Related work

Representation evaluation methods. Until recently, the standard technique for evaluating representation quality was the use of linear probes (Kiros et al., 2015; Hill et al., 2016; van den Oord et al., 2018; Chen et al., 2020). However, Hénaff et al. (2020) find that evaluation with linear probes is largely uncorrelated with the more practically relevant objective of low-data accuracy, and Resnick et al. (2019) show that linear probe performance does not predict performance for transfer across tasks. Beyond linear probes, Zhang & Bowman (2018) and Hewitt & Liang (2019) show that restrictions on model capacity or evaluation dataset size are necessary to separate the performance of randomly- and linguistically-pretrained representations. Voita & Titov (2020) propose using the MDL framework, which measures the description length of the labels given the observations. An earlier work by Yogatama et al. (2019) also uses prequential codes to evaluate representations for linguistic tasks. Talmor et al. (2019) look at the loss-data curve (called "learning curve" in their work) and use a weighted average of the validation loss at various training set sizes to evaluate representations.

Foundational work. A fundamental paper by Blier & Ollivier (2018) introduces prequential codes as a measure of the complexity of a deep learning model. Xu et al. (2020) introduce predictive $\mathcal{V}$-information, a theoretical generalization of mutual information which takes into account computational constraints, and is essentially the mutual information lower bound often reported in practice. Work by Dubois et al. (2020) describe representations which, in combination with a specified family of predictive functions, have guarantees on their generalization performance.

Appendix F Algorithmic details for estimating surplus description length

Recall that the SDL is defined as

$$m_{\mathrm{SDL}}(\phi, \mathcal{D}, \mathcal{A}, \varepsilon) = \sum_{n=1}^{\infty} \Big[L(\mathcal{A}_\phi, n) - \varepsilon \Big]_+ \tag{13}$$

For simplicity, we assume that L is bounded in $[0, 1]$. Note that this can be achieved by truncating the cross-entropy loss.

In our experiments we replace $D_M^k[1:n]$ of Algorithm 1 with sampled subsets of size n from a single evaluation dataset. Additionally, we use between 10 and 20 values of n instead of evaluating $L(\mathcal{A}_\phi, n)$ at every integer between 1 and M. This strategy, also used by Blier & Ollivier (2018) and Voita & Titov (2020), corresponds to the description length under a code which updates only periodically during transmission of the data instead of after every single point.

Theorem 4. *Let the loss function L be bounded in $[0, 1]$ and assume that it is decreasing in n. With $(M+1)K$ datapoints, if the sample complexity is less than M, the above algorithm returns an estimate $\hat{m}$ such that with probability at least $1-\delta$*

$$|\hat{m} - m(\phi, \mathcal{D}, \varepsilon, \mathcal{A})| \leq M\sqrt{\frac{\log(2M/\delta)}{2K}}. \tag{14}$$

Algorithm 1: Estimate surplus error

Input: tolerance ε, max iterations M, number of datasets K, representation ϕ, data distribution $\mathcal{D}$, algorithm $\mathcal{A}$

Output: Estimate $\hat{m}$ of $m(\phi, \mathcal{D}, \varepsilon, \mathcal{A})$ and indicator I of whether this estimate is tight or lower bound

```
Sample $K$ datasets $D_M^k \sim \mathcal{D}$ of size $M+1$
for $n = 1$ to $M$ do
    For each $k \in [K]$, run $\mathcal{A}$ on $D_M^k[1:n]$ to produce a predictor $\hat{p}_n^k$
    Take $K$ test samples $(x_k, y_k) = D_M^k[M+1]$
    Evaluate $\hat{L}_n = \frac{1}{K}\sum_{k=1}^K \ell(\hat{p}_n^k, x_k, y_k)$
Set $\hat{m} = \sum_{n=1}^M [\hat{L}_n - \varepsilon]_+$
if $\hat{L}_M \le \varepsilon/2$ then Set $I =$ tight else Set $I =$ lower bound;
return $\hat{m}, I$
```

If $K \ge \frac{\log(1/\delta)}{2\varepsilon^2}$ and the algorithm returns `tight` *then with probability at least $1-\delta$ the sample complexity is less than M and the above bound holds.*

Proof. First we apply a Hoeffding bound to show that each $\hat{L}_n$ is estimated well. For any n, we have

$$P\left(\left|\hat{L}_n - L(\mathcal{A}_\phi, n)\right| > \sqrt{\frac{\log(2M/\delta)}{2K}}\right) \le 2\exp\left(-2K\frac{\log(2M/\delta)}{2K}\right) = 2\frac{\delta}{2M} = \frac{\delta}{M} \tag{15}$$

since each $\ell(\hat{p}_n^k, x_k, y_k)$ is an independent variable, bounded in [0,1] with expectation $L(\mathcal{A}_\phi, n)$.

Now when sample complexity is less than M, we use a union bound to translate this to a high probability bound on error of $\hat{m}$, so that with probability at least $1-\delta$:

$$\begin{aligned}|\hat{m} - m(\phi, \mathcal{D}, \varepsilon, \mathcal{A})| &= \left|\sum_{n=1}^M [\hat{L}_n - \varepsilon]_+ - [L(\mathcal{A}_\phi, n) - \varepsilon]_+\right| && (16)\\ &\le \sum_{n=1}^M \left|[\hat{L}_n - \varepsilon]_+ - [L(\mathcal{A}_\phi, n) - \varepsilon]_+\right| && (17)\\ &\le \sum_{n=1}^M \left|\hat{L}_n - L(\mathcal{A}_\phi, n)\right| && (18)\\ &\le M\sqrt{\frac{\log(2M/\delta)}{2K}} && (19)\end{aligned}$$

This gives us the first part of the claim.

We want to know that when the algorithm returns `tight`, the estimate can be trusted (i.e. that we set M large enough). Under the assumption of large enough K, and by an application of Hoeffding, we have that

$$P\left(L(\mathcal{A}_\phi, M) - \hat{L}_M > \varepsilon/2\right) \le \exp\left(-2K\varepsilon^2\right) \le \exp\left(-2\frac{\log(1/\delta)}{2\varepsilon^2}\varepsilon^2\right) = \delta \tag{20}$$

If $\hat{L}_M \le \varepsilon/2$, this means that $L(\mathcal{A}_\phi, M) \le \varepsilon$ with probability at least $1-\delta$. By the assumption of decreasing loss, this means the sample complexity is less than M, so the bound on the error of $\hat{m}$ holds. □

APPENDIX G ALGORITHMIC DETAILS FOR ESTIMATING SAMPLE COMPLEXITY

Recall that ε sample complexity (εSC) is defined as

$$m_{\varepsilon\text{SC}}(\phi, \mathcal{D}, \mathcal{A}, \varepsilon) = \min\left\{n \in \mathbb{N} : L(\mathcal{A}_\phi, n) \le \varepsilon\right\}. \tag{21}$$

We estimate $m_{\varepsilon\mathrm{SC}}$ via recursive grid search. To be more precise, we first define a search interval $[1, N]$, where N is a large enough number such that $L(\mathcal{A}_\phi, N) \ll \varepsilon$. Then, we partition the search interval in to 10 sub-intervals and estimate risk of hypothesis learned from $D^n \sim \mathcal{D}^n$ with high confidence for each sub-interval. We then find the leftmost sub-interval that potentially contains $m_{\varepsilon\mathrm{SC}}$ and proceed recursively. This procedure is formalized in Algorithm 2 and its guarantee is given by Theorem 5.

Algorithm 2: Estimate sample complexity via recursive grid search

Input: Search upper limit N, parameters ε, confidence parameter δ, data distribution $\mathcal{D}$, and learning algorithm $\mathcal{A}$.

Output: Estimate $\hat{m}$ such that $m_{\varepsilon\mathrm{SC}}(\phi, \mathcal{D}, \mathcal{A}, \varepsilon) \leq \hat{m}$ with probability $1 - \delta$.

```
let $S = 2\log(20k/\delta)/\varepsilon^2$, and let $[\ell, u]$ be the search interval initialized at $\ell = 1, u = N$.
for $r = 1$ to $k$ do
    Partition $[\ell, u]$ into 10 equispaced bins and let $\Delta$ be the length of each bin.
    for $j = 1$ to $10$ do
        Set $n = \ell + j\Delta$.
        Compute $\hat{L}_n = \frac{1}{S}\sum_{i=1}^{S} \ell(\mathcal{A}(D_i^n), x_i, y_i)$ for $S$ independent draws of $D^n$ and test
          sample $(x, y)$.
        if $\hat{L}_n \leq \varepsilon/2$ then
            Set $u = n$ and $\ell = n - \Delta$.
            break
return $\hat{m} = u$, which satisfies $m_{\varepsilon\mathrm{SC}}(\phi, \mathcal{D}, \mathcal{A}, \varepsilon) \leq \hat{m}$ with probability $1 - \delta$, where the
  randomness is over independent draws of $D^n$ and test samples $(x, y)$.
```

Theorem 5. *Let the loss function L be bounded in* $[0, 1]$ *and assume that it is decreasing in* n*. Then, Algorithm 2 returns an estimate* $\hat{m}$ *that satisfies* $m_{\varepsilon\mathrm{SC}}(\phi, \mathcal{D}, \mathcal{A}, \varepsilon) \leq \hat{m}$ *with probability at least* $1 - \delta$*.*

Proof. By Hoeffding, the probability that $|\hat{L}_n - L(\mathcal{A}_\phi, n)| \geq \varepsilon/2$, where $\hat{L}$ is computed with $S = 2\log(20k/\delta)/\varepsilon^2$ independent draws of $D^n \sim \mathcal{D}^n$ and $(x, y) \sim \mathcal{D}$, is less than $\delta/(10k)$. The algorithm terminates after evaluating $\hat{L}$ on at most $10k$ different n's. By a union bound, the probability that $|\hat{L}_n - L(\mathcal{A}_\phi, n)| \leq \varepsilon/2$ for all n used by the algorithm is at least $1 - \delta$. Hence, $\hat{L}_n \leq \varepsilon/2$ implies $L(\mathcal{A}_\phi, n) \leq \varepsilon$ with probability at least $1 - \delta$. □

Appendix H Experimental details

In each experiment we first estimate the loss-data curve using a fixed number of dataset sizes n and multiple random seeds, then compute each measure from that curve. Reported values of SDL correspond to the estimated area between the loss-data curve and the line $y = \varepsilon$ using Riemann sums with the values taken from the left edge of the interval. This is the same as the chunking procedure of Voita & Titov (2020) and is equivalent to the code length of transmitting each chunk of data using a fixed model and switching models between intervals. Reported values of εSC correspond to the first measured n at which the loss is less than ε.

All of the experiments were performed on a single server with 4 NVidia Titan X GPUs, and on this hardware no experiment took longer than an hour. All of the code for our experiments, as well as that used to generate our plots and tables, is included in the supplement.

H.1 MNIST experiments

For our experiments on MNIST, we implement a highly-performant vectorized library in JAX to construct loss-data curves. With this implementation it takes about one minute to estimate the loss-data curve with one sample at each of 20 settings of n. We approximate the loss-data curves at 20 settings of n log-uniformly spaced on the interval $[10, 50000]$ and evaluate loss on the test set to approximate the population loss. At each dataset size n we perform the same number of updates to the model; we experimented with early stopping for smaller n but found that it made no difference on this dataset. In order to obtain lower-variance estimates of the expected risk at each n, we run 8 random seeds for each representation at each dataset size, where each random seed corresponds to a random initialization of the probe network and a random subsample of the evaluation dataset.

Probes consist of two-hidden-layer MLPs with hidden dimension 512 and ReLU activations. All probes and representations are trained with the Adam optimizer (Kingma & Ba, 2015) with learning rate 10^{-4}.

Each representation is normalized to have zero mean and unit variance before probing to ensure that differences in scaling and centering do not disrupt learning. The representations of the data we evaluate are implemented as follows.

Raw pixels. The raw MNIST pixels are provided by the Pytorch `datasets` library (Paszke et al., 2019). It has dimension $28 \times 28 = 784$.

CIFAR. The CIFAR representation is given by the last hidden layer of a convolutional neural network trained on the CIFAR-10 dataset. This representation has dimension 784 to match the size of the raw pixels. The network architecture is as follows:

```
    nn.Conv2d(1, 32, 3, 1),
    nn.ReLU(),
    nn.MaxPool2d(2),
    nn.Conv2d(32, 64, 3, 1),
    nn.ReLU(),
    nn.MaxPool2d(2),
    nn.Flatten(),
    nn.Linear(1600, 784)
    nn.ReLU()
    nn.Linear(784, 10)
    nn.LogSoftmax()
```

VAE. The VAE (variational autoencoder; Kingma & Welling (2014); Rezende et al. (2014)) representation is given by a variational autoencoder trained to generate the MNIST digits. This VAE's latent variable has dimension 8. We use the mean output of the encoder as the representation of the data. The network architecture is as follows:

```
self.encoder_layers = nn.Sequential(
    nn.Linear(784, 400),
    nn.ReLU(),
    nn.Linear(400, 400),
    nn.ReLU(),
    nn.Linear(400, 400),
    nn.ReLU(),
)
self.mean = nn.Linear(400, 8)
self.variance = nn.Linear(400, 8)

self.decoder_layers = nn.Sequential(
    nn.Linear(8, 400),
    nn.ReLU(),
    nn.Linear(400, 400),
    nn.ReLU(),
    nn.Linear(400, 784),
)
```

H.2 Part of speech experiments

We follow the methodology and use the official code[2] of Voita & Titov (2020) for our part of speech experiments using ELMo (Peters et al., 2018) pretrained representations. In order to obtain lower-variance estimates of the expected risk at each n, we run 4 random seeds for each representation at each dataset size, where each random seed corresponds to a random initialization of the probe network and a random subsample of the evaluation dataset. We approximate the loss-data curves at 10 settings of n log-uniformly spaced on the range of the available data $n \in [10, 10^6]$. To more precisely estimate εSC, we perform one recursive grid search step: we space 10 settings over the range which in the first round saw $L(\mathcal{A}_\phi, n)$ transition from above to below ε.

Probes consist of the MLP-2 model of Hewitt & Liang (2019); Voita & Titov (2020) and all training parameters are the same as in those works.

[2] https://github.com/lena-voita/description-length-probing