UNPACKING INFORMATION BOTTLENECKS: SURROGATE OBJECTIVES FOR DEEP LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

The Information Bottleneck principle offers both a mechanism to explain how deep neural networks train and generalize, as well as a regularized objective with which to train models. However, multiple competing objectives are proposed in the literature, and the information-theoretic quantities used in these objectives are difficult to compute for large deep neural networks, which in turn limits their use as a training objective. In this work, we review these quantities, and compare and unify previously proposed objectives, which allows us to develop surrogate objectives more friendly to optimization without relying on cumbersome tools such as density estimation. We find that these surrogate objectives allow us to apply the information bottleneck to modern neural network architectures. We demonstrate our insights on MNIST, CIFAR-10 and ImageNette with modern DNN architectures (ResNets).

1 INTRODUCTION

The Information Bottleneck (IB) principle, introduced by Tishby et al. (2000), proposes that training and generalization in deep neural networks (DNNs) can be explained by information-theoretic principles (Tishby and Zaslavsky, 2015; Shwartz-Ziv and Tishby, 2017; Achille and Soatto, 2018a). This is attractive as the success of DNNs remains largely unexplained by tools from computational learning theory (Zhang et al., 2016; Bengio et al., 2009). The IB principle suggests that learning consists of two competing objectives: maximizing the mutual information between the latent representation and the label to promote accuracy, while at the same time minimizing the mutual information between the latent representation and the input to promote generalization. Following this principle, many variations of IB objectives have been proposed (Alemi et al., 2016; Strouse and Schwab, 2017; Fisher, 2019; Gondek and Hofmann, 2003; Achille and Soatto, 2018a), which, in supervised learning, have been demonstrated to benefit robustness to adversarial attacks (Alemi et al., 2016; Fisher, 2019) and generalization and regularization against overfitting to random labels (Fisher, 2019).

Whether the benefits of training with IB *objectives* are due to the IB *principle*, or some other unrelated mechanism, remains unclear (Saxe et al., 2019; Amjad and Geiger, 2019; Tschannen et al., 2019), suggesting that although recent work has also tied the principle to successful results in both unsupervised and self-supervised learning (Oord et al., 2018; Belghazi et al., 2018; Zhang et al., 2018, among others), our understanding of how IB objectives affect representation learning remains unclear.

Critical to studying this question is the computation of the information-theoretic quantities[1] used. While progress has been made in developing mutual information estimators for DNNs (Poole et al., 2019; Belghazi et al., 2018; Noshad et al., 2019; McAllester and Stratos, 2018; Kraskov et al., 2004), current methods still face many limitations when concerned with high-dimensional random variables (McAllester and Stratos, 2018) and rely on complex estimators or generative models. This presents a challenge to training with IB objectives.

In this paper, we analyze information quantities and relate them to surrogate objectives for the IB principle which are more friendly to optimization, showing that complex or intractable IB objectives can be replaced with simple, easy-to-compute surrogates that produce similar performance and similar behaviour of information quantities over training. **Sections 2 & 3** review commonly-used information quantities for which we provide mathematically grounded intuition via information diagrams and

[1] We shorten these to *information quantities* from now on.

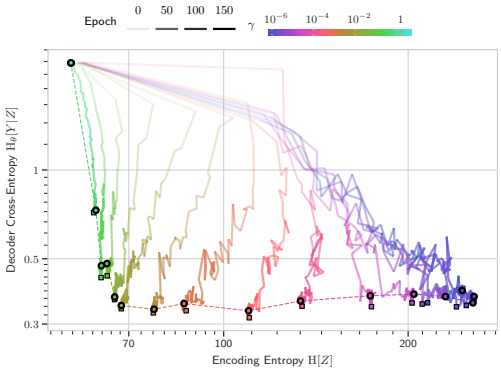


Figure 1: *Generalization plot of the training trajectories of ResNet18 models with our surrogate objective* $\min_\theta \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}] + \gamma \mathbb{E}\, \|Z\|^2$ *on* ***ImageNette***. Color shows γ; transparency the training epoch. Compression (Encoding Entropy ↓) trades-off with test performance (Decoder Cross-Entropy ↓). See section 4.

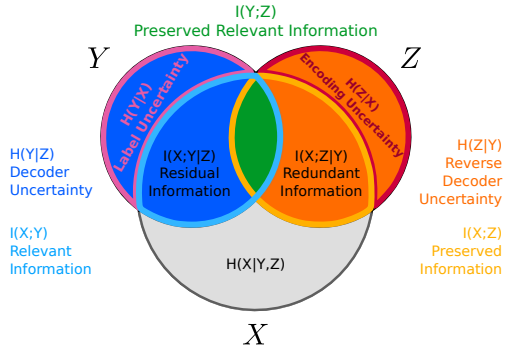


Figure 2: *Mickey Mouse I-diagram for the information quantities in the model* $\mathrm{p}(x, y, z) = \hat{\mathrm{p}}(x, y)\, \mathrm{p}_\theta(z \mid x)$. X is for the data, Y for the labels, Z for the latent encodings. See section 2 for more details and section B.1 for a description of all the quantities. $\mathrm{I}[\mathrm{Y};\mathrm{Z}] = \mathrm{I}[\mathrm{X};\mathrm{Y};\mathrm{Z}]$ because $\mathrm{I}[\mathrm{Y};\mathrm{Z} \mid \mathrm{X}] = 0$. Best viewed in color.

unify different IB objectives by identifying two key information quantities, Decoder Uncertainty $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ and Reverse Decoder Uncertainty $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$ which act as the main loss and regularization terms in our unified IB objective. In particular, **Section 3.2** demonstrates that using the Decoder Uncertainty as a training objective can minimize the training error, and shows how to estimate an upper bound on it efficiently for well-known DNN architectures. We expand on the findings of Alemi et al. (2016) in their variational IB approximation and demonstrate that this upper bound is equal to the commonly-used cross-entropy loss [2] under Dropout regularization. **Section 3.3** examines pathologies of differential entropies that hinder optimization and proposes adding Gaussian noise to force differential entropies to become non-negative, which leads to new surrogate terms to optimize the Reverse Decoder Uncertainty. Altogether this leads to simple and tractable surrogate IB objectives such as the following, which uses dropout, adds Gaussian noise over the feature vectors $f(x; \eta)$, and uses an L2 penalty over the noisy feature vectors:

$$\min_\theta \mathbb{E}_{\substack{x,y\sim\hat{\mathrm{p}}(x,y),\epsilon\sim\mathcal{N} \\ \eta\sim\text{dropout mask}}} \left[-\log \mathrm{p}(\hat{\mathrm{Y}} = y \mid z = f_\theta(x;\eta) + \epsilon) + \gamma \|f_\theta(x;\eta) + \epsilon\|_2^2 \right]. \tag{1}$$

Section 4 describes experiments that validate our insights qualitatively and quantitatively on MNIST, CIFAR-10 and ImageNette, and shows that with objectives like the one in equation (1) we obtain information plane plots (as in figure 1) similar to those predicted by Tishby and Zaslavsky (2015). Our surrogate objectives thus induce the desired behavior of IB objectives while scaling to large, high-dimensional datasets which had previously remained intractable. In particular, we present the **first successful evaluation of IB objectives on CIFAR-10 and Imagenette images**.

Compared to existing work, we show that we can optimize IB objectives for well-known DNN architectures using standard optimizers, losses and simple regularizers, without needing complex estimators or generative models. This will allow future research to make better use of IB objectives and study the IB principle more thoroughly.

2 Background

Information quantities & information diagrams. We denote entropy $\mathrm{H}[\cdot]$, joint entropy $\mathrm{H}[\cdot,\cdot]$, conditional entropy $\mathrm{H}[\cdot \mid \cdot]$, mutual information $\mathrm{I}[\cdot;\cdot]$ and Shannon's information content $h(\cdot)$ (Cover and Thomas, 2012; MacKay, 2003; Shannon, 1948). We will further require the Kullback-Leibler divergence $D_{\mathrm{KL}}(\cdot \,\|\, \cdot)$ and cross-entropy $\mathrm{H}(\cdot \,\|\, \cdot)$. The definitions can be found in section A.1. We will use differential entropies interchangeably with entropies: equalities between them are preserved in the differential setting, and inequalities will be covered in section 3.3.

Information diagrams (I-diagrams), like the one depicted in figure 2, clarify the relationship between information quantities: *similar to Venn diagrams, a quantity equals the sum of its parts in the diagram.*

[2] This connection was assumed without proof by Achille and Soatto (2018a;b).

Importantly, they offer a grounded intuition as Yeung (1991) show that we can define a signed measure μ^* such that information quantities map to abstract sets and are consistent with set operations. We provide details on how to use I-diagrams and what to watch out for in section A.2.

Probabilistic model. We will focus on a supervised classification task that makes prediction $\hat{Y}$ given data X using a latent encoding Z, while the provided target is Y. We assume categorical Y and $\hat{Y}$, and continuous X. Our probabilistic model based on these assumptions is as follows:

$$p(x, y, z, \hat{y}) = \hat{p}(x, y)\, p_\theta(z \mid x)\, p_\theta(\hat{y} \mid z). \tag{2}$$

Thus, Z and Y are independent given X, and $\hat{Y}$ is independent of X and Y given Z. The data distribution $\hat{p}(x, y)$ is only available to us as an empirical sample distribution. θ are the parameters we would like to learn. $p_\theta(z \mid x)$ is the encoder from data X to latent Z, and $p_\theta(\hat{y} \mid z)$ the decoder from latent Z to prediction $\hat{Y}$. Together, $p_\theta(z \mid x)$ and $p_\theta(\hat{y} \mid z)$ form the discriminative model $p_\theta(\hat{y} \mid x)$:

$$p_\theta(\hat{y} \mid x) = \mathbb{E}_{p_\theta(z|x)}\, p_\theta(\hat{y} \mid z). \tag{3}$$

We can derive the cross-entropy loss $H(\hat{p}(y \mid x) \,\|\, p_\theta(\hat{Y} = y \mid x))$ (Solla et al., 1988; Hinton, 1990) by minimizing the Kullback-Leibler divergence between the empirical sample distribution $\hat{p}(x, y)$ and the parameterized distribution $p_\theta(x)\, p_\theta(\hat{y} \mid x)$, where we set $p_\theta(x) = \hat{p}(x)$. See section D.1.

Mickey Mouse I-diagram. The corresponding I-diagram for X, Y, and Z is depicted in figure 2. As some of the quantities have been labelled before, we try to follow conventions and come up with consistent names otherwise. *Section B.1 provides intuitions for these quantities, and section B.2 lists all definitions and equivalences explicitly*. For categorical Z, all the quantities in the diagram are positive, which allows us to read off inequalities from the diagram: only $I[X; Y; Z]$ could be negative, but as Y and Z are independent given X, we have $I[Y; Z|X] = 0$, and $I[X; Y; Z] = I[Y; Z] - I[Y; Z|X] = I[Y; Z] \geq 0$. Section 3.3 investigates how to preserve inequalities for continuous Z.

3 Surrogate IB & DIB objectives

3.1 IB Objectives

Tishby et al. (2000) introduce the IB objective as a relaxation of a constrained optimization problem: minimize the mutual information between the input X and its latent representation Z while still accurately predicting Y from Z. An analogous objective which yields deterministic Z, the Deterministic Information Bottleneck (DIB) was proposed by Strouse and Schwab (2017). Letting β be a Lagrange multiplier, we arrive at the IB and DIB objectives:

$$\min I[X; Z] - \beta I[Y; Z] \quad \text{for IB, and} \quad \min H[Z] - \beta I[Y; Z] \text{ for DIB.} \tag{4}$$

This principle can be recast as a generalization of finding minimal sufficient statistics for the labels given the data (Shamir et al., 2010; Tishby and Zaslavsky, 2015; Fisher, 2019): it strives for minimality and sufficiency of the latent Z. Minimality is achieved by minimizing the Preserved Information $I[X; Z]$; while sufficiency is achieved by maximizing the Preserved Relevant Information $I[Y; Z]$. We defer an in-depth discussion of the IB principle to the appendix Section C.1. We discuss the several variants of IB objectives, and justify our focus on IB and DIB, in Section C.2.

The information quantities that appear in the IB objective are not tractable to compute for the representations learned by many function classes of interest, including neural networks; for example, Strouse and Schwab (2017) only obtain an analytical solution to their Deterministic Information Bottleneck (DIB) method for the tabular setting. Alemi et al. (2016) address this challenge by constructing a variational approximation of the IB objective, but their approach has not been applied to more complex datasets than MNIST variants. Belghazi et al. (2018) use a separate statistics network to approximate the mutual information, a computationally expensive strategy that does not easily lend itself to optimization.

In this section, we introduce and justify tractable surrogate losses that are easier to apply in common deep learning pipelines, and which can be scaled to large and high-dimensional datasets. We begin by proposing the following reformulation of IB and DIB objectives.

Proposition 1. *For IB, we obtain*

$$\arg\min I[X; Z] - \beta I[Y; Z] = \arg\min H[Y \mid Z] + \beta' \underbrace{I[X; Z \mid Y]}_{=H[Z|Y]-H[Z|X]}, \tag{5}$$

and, for DIB,

$$\arg\min \mathrm{H}[Z] - \beta \mathrm{I}[Y;Z] = \arg\min \mathrm{H}[Y \mid Z] + \beta' \mathrm{H}[Z \mid Y] = \arg\min \mathrm{H}[Y \mid Z] + \beta'' \mathrm{H}[Z] \tag{6}$$

with $\beta' := \frac{1}{\beta - 1} \in [0, \infty)$ *and* $\beta'' := \frac{1}{\beta} \in [0, 1)$. *The derivation can be found in section C.3.*

In the next sections, we show that Decoder Uncertainty $\mathrm{H}[Y \mid Z]$ provides a loss term, which minimizes the training error, and DIB's Reverse Decoder Uncertainty $\mathrm{H}[Z \mid Y]$ and IB's Redundant Information $\mathrm{I}[X;Z \mid Y]$, respectively, provide a regularization term, which helps generalization. Another perspective can be found by relating the objectives to the Entropy Distance Metric introduced by MacKay (2003), which we detail in section C.4.

3.2 Decoder Uncertainty $\mathrm{H}[Y \mid Z]$

The Decoder Uncertainty $\mathrm{H}[Y \mid Z]$ is the first term in our reformulated IB and DIB objectives, and captures the data fit component of the IB principle. This quantity is not easy to compute directly for arbitrary representations Z, so we turn our attention to two related entities instead, where we use θ as subscript to mark dependence on the model: the Prediction Cross-Entropy, denoted $\mathrm{H}_\theta[Y \mid X]$ (more commonly known as the model's cross-entropy loss; see section D.1), and the Decoder Cross-Entropy, denoted $\mathrm{H}_\theta[Y \mid Z]$. Noting that $h(x) = -\ln x$, we define these terms as follows:

$$\mathrm{H}_\theta[Y \mid X] := \mathrm{H}(\hat{\mathrm{p}}(y \mid x) \,\|\, \mathrm{p}_\theta(\hat{Y} = y \mid x)) = \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, h\Big(\mathbb{E}_{\mathrm{p}_\theta(z|x)}\, \mathrm{p}_\theta(\hat{Y} = y \mid z)\Big) \tag{7}$$

$$\mathrm{H}_\theta[Y \mid Z] := \mathrm{H}(\mathrm{p}(y \mid z) \,\|\, \mathrm{p}_\theta(\hat{Y} = y \mid z)) = \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, \mathbb{E}_{\mathrm{p}_\theta(z|x)}\, h\Big(\mathrm{p}_\theta(\hat{Y} = y \mid z)\Big). \tag{8}$$

Jensen's inequality yields $\mathrm{H}_\theta[Y \mid X] \leq \mathrm{H}_\theta[Y \mid Z]$, with equality iff Z is a deterministic function of X. The notational similarity[3] between $\mathrm{H}_\theta[Y \mid Z]$ and $\mathrm{H}[Y \mid Z]$ is deliberately suggestive: this cross-entropy bounds the conditional entropy $\mathrm{H}[Y \mid Z]$, as characterized in the following proposition.

Proposition 2. *The Decoder Cross-Entropy provides an upper bound on the Decoder Uncertainty:*

$$\mathrm{H}[Y \mid Z] \leq \mathrm{H}[Y \mid Z] + D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \,\|\, \mathrm{p}_\theta(\hat{y} \mid z)) = \mathrm{H}_\theta[Y \mid Z], \tag{9}$$

and further bounds the training error:

$$\mathrm{p}(\text{“}\hat{Y} \textit{ is wrong}\text{”}) \leq 1 - e^{-\mathrm{H}_\theta[Y|Z]} = 1 - e^{-\left(\mathrm{H}[Y|Z] + D_{\mathrm{KL}}(\mathrm{p}(y|z) \| \mathrm{p}_\theta(\hat{y}|z))\right)}. \tag{10}$$

Likewise, for $\mathrm{H}_\theta[Y \mid X]$ *and* $\mathrm{H}[Y \mid X]$. *See section D.2 for a derivation.*

Hence, by bounding $D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \,\|\, \mathrm{p}_\theta(\hat{y} \mid z))$, we can obtain a bound for the training error in terms of $\mathrm{H}[Y \mid Z]$. We examine one way of doing so by using optimal decoders $\mathrm{p}_\theta(\hat{y} \mid z) := \mathrm{p}(Y = \hat{y} \mid z)$ for the case of categorical Z in section E.

Alemi et al. (2016) use the Decoder Cross-Entropy bound in equation (9) to variationally approximate $\mathrm{p}(y \mid z)$. We make this explicit by applying the reparameterization trick to rewrite the latent z as a parametric function of its input x and some independent auxiliary random variable η, i.e. $f_\theta(x, \eta) \overset{D}{=} z \sim \mathrm{p}_\theta(z \mid x)$, yielding

$$\mathrm{H}[Y \mid Z] \leq \mathrm{H}_\theta[Y \mid Z] = \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, \mathbb{E}_{\mathrm{p}(\eta)}\, h\Big(\mathrm{p}_\theta(\hat{Y} = y \mid z = f_\theta(x;\eta))\Big). \tag{11}$$

Equation (11) can be applied to many forms of stochastic regularization that turn deterministic models into stochastic ones, in particular Dropout. This allows us to use modern DNN architectures as stochastic encoders.

Dropout regularization When we interpret η as a sampled Dropout mask for a DNN, DNNs that use Dropout regularization (Srivastava et al., 2014), or variants like DropConnect (Wan et al., 2013a), fit the equation above as stochastic encoders. Monte-Carlo Dropout (Gal and Ghahramani, 2016), for example, even specifically estimates the predictive mean $\mathrm{p}_\theta(\hat{y} \mid x)$ from equation (3). The following result extends the observation by Burda et al. (2015) that sampling yields an unbiased estimator for the Decoder Cross-Entropy $\mathrm{H}_\theta[Y \mid Z]$, while it only yields a biased estimator for the Prediction Cross-Entropy $\mathrm{H}_\theta[Y \mid X]$ (which it upper-bounds).

Corollary 1. *Let* x, y, z *and* f_θ *be defined as previously, with* η *a sampled stochastic dropout mask. Then* $h\Big(\mathrm{p}_\theta(\hat{Y} = y \mid z = f_\theta(x;\eta))\Big)$ *evaluated for a single sample* η *is an unbiased estimator of the Decoder Cross-Entropy* $\mathrm{H}_\theta[Y \mid Z]$, *and an estimator of an upper bound on the Prediction Cross-Entropy* $\mathrm{H}_\theta[Y \mid X]$.

[3] This notation is compatible with $\mathcal{V}$-Entropy introduced by Xu et al. (2020).

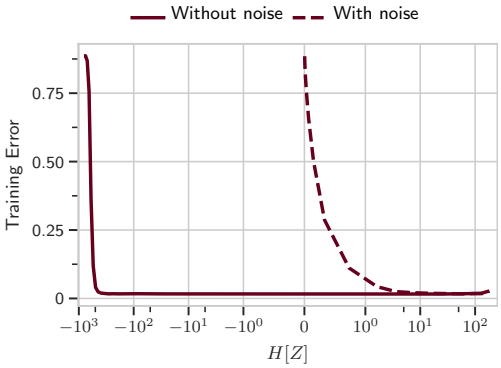


Figure 3: *Decreasing the entropy of a noise-free latent does not affect the training error.* (Though floating-point issues start affecting it negatively eventually.) When adding zero-entropy noise, the error rate increases as the entropy approaches zero. See section 3.3 and G.4 for more details.

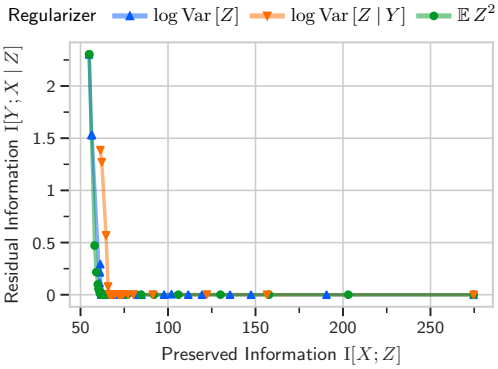


Figure 4: *Information plane plot of the latent* Z *similar to Tishby and Zaslavsky (2015) but using a* ResNet18 *model on* CIFAR-10 *using the different regularizes from section 3.3.* A larger version can be found in figure G.1. See section 4 for more details. Best viewed in color.

This distinction between the Decoder Cross-Entropy and the Prediction Cross-Entropy has been observed in passing in the literature, but not made explicit. Multi-sample approaches like Multi-Sample Dropout (Inoue, 2019), for example, optimize $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$, while Importance Weighted Stochastic Gradient Descent (Noh et al., 2017) optimizes $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}]$. Dusenberry et al. (2020) observe empirically in the different context of rank-1 Bayesian Neural Networks that optimizing $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$ instead of $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}]$ is both easier and also yields better generalization performance (NLL, accuracy, and ECE), while they also put forward an argument for why the stochastic gradients for $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$ might benefit from lower variance. We conclude this section by highlighting that $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ is therefore already minimized in modern DNN architectures that use Dropout together with a cross-entropy loss. This means that, at least for one half of our reformulation of the IB objective, we can apply off-the-shelf, scalable objectives and optimizers for its minimization.

3.3 Surrogates for the regularization terms

In the previous section, we have examined how to tractably estimate the error minimization term $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$. In this section, we will examine tractable optimization of the regularization terms $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$ and $\mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}]$, respectively. We discuss how to minimize entropies meaningfully and show how this unifies DIB and IB via the inequality $\mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}] \leq \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] \leq \mathrm{H}[\mathrm{Z}]$ before providing tractable upper-bounds for $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$ and $\mathrm{H}[\mathrm{Z}]$.

Differential entropies In most cases, the latent Z is a continuous random variable in many dimensions. Unlike entropies on discrete probability spaces, differential entropies defined on continuous spaces are not bounded from below. This means that the DIB objective is not guaranteed to have an optimal solution and allows for pathological optimization trajectories in which the variance of the latent Z can be scaled to be arbitrarily small, achieving arbitrarily high-magnitude negative entropy. We provide a toy experiment demonstrating this in section G.4.

Intuitively, one can interpret this issue as being allowed to encode information in an arbitrarily-small real number using infinite precision, similar to arithmetic coding (MacKay, 2003; Shwartz-Ziv and Tishby, 2017)[4]. In practice, due to floating point constraints, optimizing DIB naively will invariably end in garbage predictions and underflow as activations approach zero. It is therefore not desirable for training. This is why Strouse and Schwab (2017) only consider analytical solutions to DIB by evaluating a limit for the tabular case. MacKay (2003) proposes the introduction of noise to solve this issue in the application of continuous communication channels.

However, here we propose adding specific noise to the latent representation to lower-bound the conditional entropy of Z, which allows us to enforce non-negativity across all IB information quantities as in the discrete case and transport inequalities to the continuous case: for a continuous

[4]Conversely, MacKay (2003) notes that without upper-bounding the "power" $\mathbb{E}_{p(z)}\, \mathrm{Z}^2$, all information could be encoded in a single very large integer.

$\hat{Z} \in \mathbb{R}^k$ and independent noise ϵ, we set $Z := \hat{Z} + \epsilon$; the differential entropy then satisfies $H[Z] = H[\hat{Z} + \epsilon] \geq H[\epsilon]$; and by using *zero-entropy noise* $\epsilon \sim \mathcal{N}(0, \frac{1}{2\pi e} I_k)$ specifically, we obtain $H[Z] \geq H[\epsilon] = 0$.

Proposition 3. *After adding zero-entropy noise, the inequality* $I[X; Z \mid Y] \leq H[Z \mid Y] \leq H[Z]$ *also holds for continuous* Z*, and we can minimize* $I[X; Z \mid Y]$ *in the IB objective by minimizing* $H[Z \mid Y]$ *or* $H[Z]$*, similarly to the DIB objective.*

Strictly speaking, zero-entropy noise is not necessary for optimizing the bounds: any Gaussian noise is sufficient, but zero-entropy noise is aesthetically appealing as it preserves inequalities from the discrete setting. In a sense, this propostion bounds the IB objective by the DIB objective. However, adding noise changes the optimal solutions: whereas DIB in Strouse and Schwab (2017) leads to hard clustering in the limit, adding noise leads to soft clustering when optimizing the DIB objective, as is the case with the IB objective. We show in section F.4 that minimizing the DIB objective with noise leads to soft clustering (for the case of an otherwise deterministic encoder). Altogether, in addition to Shwartz-Ziv and Tishby (2017), we argue that noise is essential to obtain meaningful differential entropies and to avoid other pathological cases as described further in section F.5.

It is not generally possible to compute $H[Z \mid Y]$ exactly for continuous latent representations Z, but we can derive an upper bound. The maximum-entropy distribution for a given covariance matrix Σ is a Gaussian with the same covariance.

Proposition 4. *The Reverse Decoder Uncertainty can be approximately bounded using the empirical variance* $\widehat{\mathrm{Var}}[Z_i \mid y]$*:*

$$H[Z \mid Y] \leq \mathbb{E}_{\hat{p}(y)} \sum_i \tfrac{1}{2} \ln(2\pi e \, \mathrm{Var}[Z_i \mid y]) \approx \mathbb{E}_{\hat{p}(y)} \sum_i \tfrac{1}{2} \ln(2\pi e \, \widehat{\mathrm{Var}}[Z_i \mid y]), \tag{12}$$

where Z_i *are the individual components of* Z*.* $H[Z]$ *can be bounded similarly. More generally, we can create an even looser upper bound by bounding the mean squared norm of the latent:*

$$\mathbb{E}\, \|Z\|^2 \leq C' \Rightarrow H[Z \mid Y] \leq H[Z] \leq C, \tag{13}$$

with $C' := \frac{k e^{2C/k}}{2\pi e}$ *for* $Z \in \mathbb{R}^k$*. See section F.2 for proof.*

Surrogate objectives These surrogate terms provide us with three different upper-bounds that we can use as surrogate regularizers. We refer to them as: conditional log-variance regularizer ($\log \mathrm{Var}[Z \mid Y]$), log-variance regularizer ($\log \mathrm{Var}[Z]$) and activation L_2 regularizer ($\mathbb{E}\, \|Z\|^2$). We can now propose the main results of this paper: IB surrogate objectives that reduce to an almost trivial implementation using the cross-entropy loss and one of the regularizers above while adding zero-entropy noise to the latent Z.

Theorem 1. *Let* Z *be obtained by adding a single sample of zero-entropy noise to a single sample of the output* z *of the stochastic encoder. Then each of the following objectives is an estimator of an upper bound on the IB objective. In particular, for the surrogate objective* $\mathbb{E}\, \|Z\|^2$*, we obtain:*

$$\min H(p(y \mid z) \,\|\, p_\theta(\hat{Y} = y \mid z)) + \gamma \|z\|^2; \tag{14}$$

for $\log \mathrm{Var}[Z \mid Y]$*:*

$$\min H(p(y \mid z) \,\|\, p_\theta(\hat{Y} = y \mid z)) + \gamma \, \mathbb{E}_{\hat{p}(y)} \sum_i \frac{1}{2} \ln(2\pi e \, \widehat{\mathrm{Var}}[Z_i \mid y]); \tag{15}$$

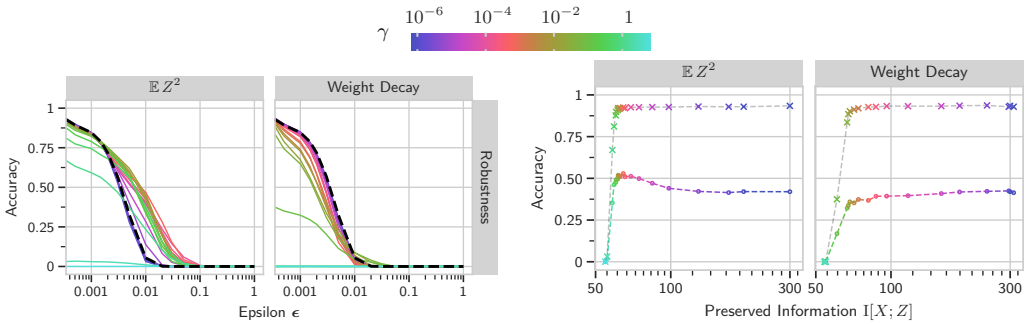
and for $\log \mathrm{Var}[Z]$*:*

$$\min H(p(y \mid z) \,\|\, p_\theta(\hat{Y} = y \mid z)) + \gamma \sum_i \frac{1}{2} \ln(2\pi e \, \widehat{\mathrm{Var}}[Z_i]). \tag{16}$$

For the latter two surrogate regularizers, we can relate their coefficient γ to β', β'' and β from section 3. However, as regularizing $\mathbb{E}\, \|Z\|^2$ does not approximate an entropy directly, its coefficient does not relate to the Lagrange multiplier of any fixed IB objective. We compare the performance of these objectives in section 4.

4 Experiments

We now provide empirical verification of the claims made in the previous sections. Our goal in this section is to highlight two main findings: first, that our surrogate objectives obtain similar behavior to what we expect of exact IB objectives with respect to their effect on robustness to adversarial examples.



(a) *Robustness for different attack strengths* ϵ. The dashed black line represents a model trained only with cross-entropy and no noise injection. We see that models trained with the surrogate IB objective (colored by γ) see improved robustness over a model trained only to minimize the cross-entropy training objective (shown in black) while the models regularized with weight-decay actually perform worse.

(b) *Average robustness over* $\epsilon \in [0, 0.1]$ *compared to normal accuracy for different amounts of Preserved Information.* $\circ$ markers show robustness. $\times$ markers show the normal accuracy. We see that robustness depends on the Preserved Information. If the latent is compressesed too much, robustness (and accuracy) are low. If the latent is not compressed enough, robustness and thus generalization suffer.

Figure 5: *Adversarial robustness of ResNet18 models trained on CIFAR-10 with surrogate objectives in comparison to regularization with L2 weight-decay as non-IB method.* The robustness is evaluated using FGSM, PGD, DeepFool and BasicIterative attacks of varying ϵ values.

In particular, we show that our surrogate IB objectives improve adversarial robustness compared to models trained only on the cross-entropy loss, consistent with the findings of Alemi et al. (2016). Second, we show the effect of our surrogate objectives on information quantities during training by plotting information plane diagrams, demonstrating that models trained with our objectives trade off between $I[X; Z]$ and $I[Y; Z]$ as expected. We show this by recovering information plane plots similar to the ones in Tishby and Zaslavsky (2015) and qualitatively examine the optimization behavior of the networks through their training trajectories. We demonstrate the scalability of our surrogate objectives by performing the first application of IB objectives to the CIFAR-10 and ImageNette datasets, high-dimensional image datasets.

For details about our experiment setup, DNN architectures, hyperparameters and additional insights, see section G. In particular, empirical quantification of our observations on the relationship between the Decoder Cross-Entropy loss and the Prediction Cross-Entropy are deferred to the appendix due to space limitations as well as the description of the toy experiment that shows that minimizing H[Z | Y] for continuous latent Z without adding noise does not constrain information meaningfully and that adding noise solves the issue as detailed in section 3.3.

Robustness to adversarial attacks Alemi et al. (2016) observe that the DVIB objective leads to improved adversarial robustness over standard training objectives. We perform a similar evaluation to see whether our surrogate objectives also see improved robustness. We train a fully-connected residual network on CIFAR-10 for a range of regularization coefficients γ using our $\mathbb{E}\,||Z||^2$ surrogate objective; we then compare against a similar regularization method that does not have an information-theoretic interpretation: L2 weight-decay. We inject zero-entropy noise in both cases. After training, we evaluate the models on adversarially perturbed images using the FGSM (Szegedy et al., 2013), PGD (Madry et al., 2018), BasicIterative (Kurakin et al., 2017) and DeepFool (Moosavi-Dezfooli et al., 2016) attacks for varying levels of the perturbation magnitude parameter ϵ. We also compare to a simple unregularized cross-entropy baseline (black dashed line). To compute overall robustness, we use each attack in turn and only count a sample as robust if it defeats them all. As depicted in figure 5, we find that our surrogate objectives yield significantly more robust models while obtaining similar test accuracy on the unperturbed data whereas weight-decay regularization reduces robustness against adversarial attacks. Plots for the other two regularizers can be found in the appendix in figure G.13 and figure G.14.

Information plane plots for CIFAR-10 To compare the different surrogate regularizers, we again use a ResNet18 model on CIFAR-10 with zero-entropy noise added to the final layer activations Z, with $K = 256$ dimensions, as an encoder and add a single $K \times 10$ linear unit as a decoder. We train with the surrogate objectives from section 3.3 for various γ, chosen in logspace from different

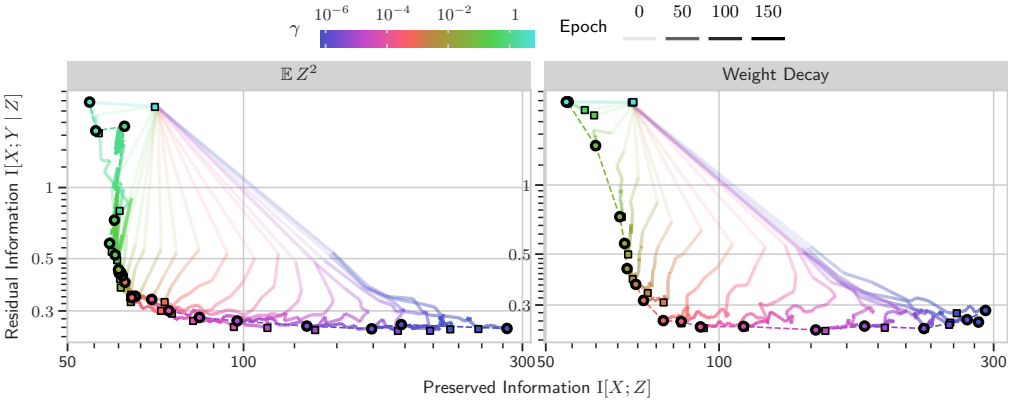


Figure 6: *Information plane plot of the training trajectories of ResNet18 models with the* $\mathbb{E}\,\|Z\|^2$ *surrogate objective or L2 weight-decay on CIFAR-10.* The color shows γ; the transparency the training epoch. Compression (Preserved Information ↓) trades-off with performance (Residual Information ↓). See section 4. While the trajectories are similar, robustness is very different, see figure 5.

ranges to compensate for their relationship to β as noted in section 3.3: for log Var[Z], $\gamma \in [10^{-5}, 1]$; for log Var[Z | Y], $\gamma \in [10^{-5}, 10]$; and for $\mathbb{E}\,\|Z\|^2$, by trial and error, $\gamma \in [10^{-6}, 10]$. We estimate information quantities using the method of Kraskov et al. (2004).

Figure 6 shows an information plane plot for regularizing with $\mathbb{E}\,\|Z\|^2$ for different γ over different epochs for the training set. Similar to Shwartz-Ziv and Tishby (2017), we observe that there is an initial expansion phase followed by compression. The jumps in performance (reduction of the Residual Information) are due to drops in the learning rate. In figure 4, we can see that the saturation curves for all 3 surrogate objectives qualitatively match the predicted curve from Tishby and Zaslavsky (2015). Figure G.1 shows the difference between the regularizers more clearly, and figure G.3 shows the training trajectories for all three regularizers.

Information plane plots for ImageNette To show that our surrogate objectives also scale up to larger datasets, we run a similar experiment on ImageNette (Howard, 2019), which is a subset of ImageNet with 10 classes with $224 \times 224 \times 3 = 1.5 \times 10^5$ input dimensions, and on which we obtain 90% test accuracy. See the figure 1, which shows the trajectories on the test set. We obtain similar plots to the ones obtained for CIFAR-10, *showing that our surrogate objectives scale well to higher-dimensional datasets.* **This is the first time that IB objectives have been applied to such large models and higher-dimensional datasets**. DVIB only experiments with 1536-dim embeddings extracted from ImageNet using pretrained models and uses 3-layer MLPs with at most 3.6×10^6 parameters. Our ResNet18 models have a complex architecture of 20 layers and 11×10^6 parameters.

5 Conclusion

The contributions of this paper have been threefold: first, we have provided mathematically grounded intuition by using I-diagrams for the information quantities involved in IB, shown common pitfalls when using information quantities and how to avoid them, and examined how the quantities relate to each other. Second, we have endeavoured to provide a unifying view on IB approaches and further provided insight into limitations of IB training, demonstrating how to avoid pathological behavior in IB objectives. Third, we have proposed tractable training objectives based on this intuition, which capture many of the desirable properties of IB methods while also scaling to problems of interest in deep learning. Specifically, we have shown how the Decoder Uncertainty $\mathrm{H}_\theta[Y \mid Z]$ relates to the cross-entropy loss that is commonly used for classification problems, and demonstrated that with these new objectives we can move beyond toy examples using IB methods to more practical DNNs without cumbersome density models. Future work investigating the practical constraints on the expressivity of a given neural network may provide further insight into how to measure compression in neural networks, and potentially yield improved regularizers over those presented in this paper. Moreover, the connection to Bayesian Neural Networks remains to be explored.

References

Alessandro Achille and Stefano Soatto. Emergence of invariance and disentanglement in deep representations. *The Journal of Machine Learning Research*, 19(1):1947–1980, 2018a.

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40 (12):2897–2905, 2018b.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Rana Ali Amjad and Bernhard Claus Geiger. Learning representations for neural network-based classification using the information bottleneck principle. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning*, pages 531–540, 2018.

Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

J-F Bercher and Christophe Vignat. A renyi entropy convolution inequality with application. In *2002 11th European Signal Processing Conference*, pages 1–4. IEEE, 2002.

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Michael W Dusenberry, Ghassen Jerfel, Yeming Wen, Yi-an Ma, Jasper Snoek, Katherine Heller, Balaji Lakshminarayanan, and Dustin Tran. Efficient and scalable bayesian neural nets with rank-1 factors. *arXiv preprint arXiv:2005.07186*, 2020.

Ian Fisher. The Conditional Entropy Bottleneck. *Submission to ICLR 2019, International Conference on Learning Representations*, 2019.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059, 2016.

David Gondek and Thomas Hofmann. Conditional information bottleneck clustering. In *3rd ieee international conference on data mining, workshop on clustering large data sets*, pages 36–42. Citeseer, 2003.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. *Lecture Notes in Computer Science*, page 630–645, 2016b.

Geoffrey E Hinton. Connectionist learning procedures. In *Machine learning*, pages 555–610. Elsevier, 1990.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

Jeremy Howard. Imagewang, 2019. URL https://github.com/fastai/imagenette/.

Hiroshi Inoue. Multi-sample dropout for accelerated training and better generalization. *arXiv preprint arXiv:1905.09788*, 2019.

Morris A. Jette, Andy B. Yoo, and Mark Grondona. Slurm: Simple linux utility for resource management. In *In Lecture Notes in Computer Science: Proceedings of Job Scheduling Strategies for Parallel Processing (JSSPP) 2003*, pages 44–60. Springer-Verlag, 2002.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems*, pages 7024–7035, 2019.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial machine learning at scale. 2017.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

David J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. *arXiv preprint arXiv:1811.04251*, 2018.

William McGill. Multivariate information transmission. *Transactions of the IRE Professional Group on Information Theory*, 4(4):93–111, 1954.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

Hyeonwoo Noh, Tackgeun You, Jonghwan Mun, and Bohyung Han. Regularizing deep neural networks by noise: Its interpretation and optimization. In *Advances in Neural Information Processing Systems*, pages 5109–5118, 2017.

Morteza Noshad, Yu Zeng, and Alfred O Hero. Scalable mutual information estimation using dependence graphs. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2962–2966. IEEE, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

Ben Poole, Sherjil Ozair, Aaron van den Oord, Alexander A Alemi, and George Tucker. On variational bounds of mutual information. *arXiv preprint arXiv:1905.06922*, 2019.

Andrew M Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan D Tracey, and David D Cox. On the information bottleneck theory of deep learning. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124020, 2019.

Ohad Shamir, Sivan Sabato, and Naftali Tishby. Learning and generalization with the information bottleneck. *Theoretical Computer Science*, 411(29-30):2696–2711, 2010.

Claude E Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3): 379–423, 1948.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Sara A. Solla, Esther Levin, and Michael Fleisher. Accelerated learning in layered neural networks. *Complex Systems*, 2, 1988.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

DJ Strouse and David J Schwab. The deterministic information bottleneck. *Neural computation*, 29 (6):1611–1630, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *arXiv preprint arXiv:1907.13625*, 2019.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013a.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066, 2013b.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints, 2020.

Raymond W Yeung. A new outlook on shannon's information measures. *IEEE transactions on information theory*, 37(3):466–474, 1991.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.

A Information quantities & information diagrams

Here we introduce notation and terminology in greater detail than in the main paper. We review well-known information quantities and provide more details on using information diagrams (Yeung, 1991).

A.1 Information quantities

We denote entropy $\mathrm{H}[\cdot]$, joint entropy $\mathrm{H}[\cdot,\cdot]$, conditional entropy $\mathrm{H}[\cdot \mid \cdot]$, mutual information $\mathrm{I}[\cdot;\cdot]$ and Shannon's information content $h(\cdot)$ following Cover and Thomas (2012); MacKay (2003); Shannon (1948) :

$$\begin{aligned}
h(x) &= -\ln x \\
\mathrm{H}[X] &= \mathbb{E}_{\mathrm{p}(x)}\, h(\mathrm{p}(x)) \\
\mathrm{H}[X,Y] &= \mathbb{E}_{\mathrm{p}(x,y)}\, h(\mathrm{p}(x,y)) \\
\mathrm{H}[X \mid Y] &= \mathrm{H}[X,Y] - \mathrm{H}[Y] \\
&= \mathbb{E}_{\mathrm{p}(y)}\, \mathrm{H}[X \mid y] = \mathbb{E}_{\mathrm{p}(x,y)}\, h(\mathrm{p}(x \mid y)) \\
\mathrm{I}[X;Y] &= \mathrm{H}[X] + \mathrm{H}[Y] - \mathrm{H}[X,Y] \\
&= \mathbb{E}_{\mathrm{p}(x,y)}\, h\left(\tfrac{\mathrm{p}(x)\,\mathrm{p}(y)}{\mathrm{p}(x,y)}\right) \\
\mathrm{I}[X;Y \mid Z] &= \mathrm{H}[X \mid Z] + \mathrm{H}[Y \mid Z] - \mathrm{H}[X,Y \mid Z],
\end{aligned}$$

where X, Y, Z are random variables and x, y, z are outcomes these random variables can take.

We use differential entropies interchangeably with entropies. We can do so because equalities between them hold as can be verified by symbolic expansions. For example,

$$\begin{aligned}
&\mathrm{H}[X,Y] = \mathrm{H}[X \mid Y] + \mathrm{H}[Y] \\
\Leftrightarrow\ &\mathbb{E}_{\mathrm{p}(x,y)}\, h(\mathrm{p}(x,y)) = \mathbb{E}_{\mathrm{p}(x,y)}\left[h(\mathrm{p}(x \mid y)) + h(\mathrm{p}(y))\right] = \mathbb{E}_{\mathrm{p}(x,y)}\left[h(\mathrm{p}(x \mid y))\right] + \mathbb{E}_{\mathrm{p}(y)}\, h(\mathrm{p}(y)),
\end{aligned}$$

which is valid in both the discrete and continuous case (if the integrals all exist). The question of how to transfer inequalities in the discrete case to the continuous case is dealt with in section 3.3.

We will further require the Kullback-Leibler divergence $D_{\mathrm{KL}}(\cdot \,\|\, \cdot)$ and cross-entropy $\mathrm{H}(\cdot \,\|\, \cdot)$:

$$\begin{aligned}
\mathrm{H}(\mathrm{p}(x) \,\|\, \mathrm{q}(x)) &= \mathbb{E}_{\mathrm{p}(x)}\, h(\mathrm{q}(x)) \\
D_{\mathrm{KL}}(\mathrm{p}(x) \,\|\, \mathrm{q}(x)) &= \mathbb{E}_{\mathrm{p}(x)}\, h\left(\tfrac{\mathrm{q}(x)}{\mathrm{p}(x)}\right) \\
\mathrm{H}(\mathrm{p}(y \mid x) \,\|\, \mathrm{q}(y \mid x)) &= \mathbb{E}_{\mathrm{p}(x)}\, \mathbb{E}_{\mathrm{p}(y|x)}\, h(\mathrm{q}(y \mid x)) \\
&= \mathbb{E}_{\mathrm{p}(x,y)}\, h(\mathrm{q}(y \mid x)) \\
D_{\mathrm{KL}}(\mathrm{p}(y \mid x) \,\|\, \mathrm{q}(y \mid x)) &= \mathbb{E}_{\mathrm{p}(x,y)}\, h\left(\tfrac{\mathrm{q}(y|x)}{\mathrm{p}(y|x)}\right)
\end{aligned}$$

A.2 Information diagrams

Information diagrams (I-diagrams), like the one depicted in figure 2 (or figure H.1 for a bigger version), visualize the relationship between information quantities: Yeung (1991) shows that we can define a signed measure μ^* such that these well-known quantities map to abstract sets and are consistent with set operations.

$$\begin{aligned}
\mathrm{H}[A] &= \mu^*(A) \\
\mathrm{H}[A_1,\ldots,A_n] &= \mu^*(\cup_i A_i) \\
\mathrm{H}[A_1,\ldots,A_n \mid B_1,\ldots,B_n] &= \mu^*(\cup_i A_i - \cup_i B_i) \\
\mathrm{I}[A_1;\ldots;A_n] &= \mu^*(\cap_i A_i) \\
\mathrm{I}[A_1;\ldots;A_n \mid B_1,\ldots,B_n] &= \mu^*(\cap_i A_i - \cup_i B_i)
\end{aligned}$$

Note that interaction information (McGill, 1954) follows as canonical generalization of the mutual information to multiple variables from that work, whereas total correlation does not.

In other words, equalities can be read off directly from I-diagrams: an information quantity is the sum of its parts in the corresponding I-diagram. This is similar to Venn diagrams. The sets used in I-diagrams are just abstract symbolic objects, however.

An important distinction between I-diagrams and Venn diagrams is that while we can always read off inequalities in Venn diagrams, this is not true for I-diagrams in general because mutual information terms in more than two variables can be negative. In Venn diagrams, a set is always larger or equal any subset.

However, if we show that all information quantities are non-negative, we can read off inequalities again. We do this for figure 2 at the end of section 2 for categorical Z and expand this to continuous Z in section 3.3. Thus, we can treat the Mickey Mouse I-diagram like a Venn diagram to read off equalities and inequalities.

Nevertheless, caution is warranted sometimes. As the signed measure can be negative, $\mu^*(X \cap Y) = 0$ does *not* imply $X \cap Y = \emptyset$: deducing that a mutual information term is 0 does not imply that one can simply remove the corresponding area in the I-diagram. There could be Z with $\mu^*((X \cap Y) \cap Z) < 0$, such that $\mu^*(X \cap Y) = \mu^*(X \cap Y \cap Z) + \mu^*(X \cap Y - Z) = 0$ but $X \cap Y \neq \emptyset$. This also means that we cannot drop the term from expressions when performing symbolic manipulations. This is of particular importance because a mutual information of zero means two random variables are independent, which might invite one drawing them as disjoint areas.

The only time where one can safely remove an area from the diagram is for *atomic* quantities, which are quantities which reference all the available random variables (Yeung, 1991). For example, when we only have three variables X, Y, Z, $\mathrm{I}[X; Y; Z]$ and $\mathrm{I}[X; Y \mid Z]$ are atomic quantities. We can safely remove atomic quantities from I-diagrams when they are 0 as there are no random variables left to apply that could lead to the problem explored above.

Continuing the example, $0 = \mathrm{I}[X; Y; Z] = \mu^*(X \cap Y \cap Z)$ would imply $X \cap Y \cap Z = \emptyset$, and we could remove it from the diagram without loss of generality. Moreover, atomic $\mathrm{I}[X; Y|Z] = \mu^*(X \cap Y - Z) = 0$ then and could be removed from the diagram as well.

We only use I-diagrams for the three variable case, but they supply us with tools to easily come up with equalities and inequalities for information quantities. In the general case with multiple variables, they can be difficult to draw, but for Markov chains they can be of great use.

B Mickey Mouse I-diagram

B.1 Intuition for the Mickey Mouse information quantities

We base the names of information quantities on existing conventions and come up with sensible extensions. For example, the name *PreservedRelevantInformation* for $\mathrm{I}[Y; Z]$ was introduced by Tishby and Zaslavsky (2015). It can be seen as the intersection of $\mathrm{I}[X; Z]$ and $\mathrm{I}[X; Y]$ in the I-diagram, and hence we denote $\mathrm{I}[X; Z]$ Preserved Information and $\mathrm{I}[X; Y]$ Relevant Information, which are sensible names as we detail below.

We identify the following six atomic quantities:

Label Uncertainty $\mathrm{H}[Y \mid X]$ quantifies the uncertainty in our labels. If we have multiple labels for the same data sample, it will be > 0. It is 0 otherwise.

Encoding Uncertainty $\mathrm{H}[Z \mid X]$ quantifies the uncertainty in our latent encoding given a sample. When using a Bayesian model with random variable ω for the weights, one can further split this term into $\mathrm{H}[Z \mid X] = \mathrm{I}[Z; \omega \mid X] + \mathrm{H}[Z \mid X, \omega]$, so uncertainty stemming from weight uncertainty and independent noise (Houlsby et al., 2011; Kirsch et al., 2019).

Preserved Relevant Information $\mathrm{I}[Y; Z]$ quantifies information in the latent that is relevant for our task of predicting the labels (Tishby and Zaslavsky, 2015). Intuitively, we want to maximize it for good predictive performance.

Residual Information $\mathrm{I}[X; Y \mid Z]$ quantifies information for the labels that is not captured by the latent (Tishby and Zaslavsky, 2015) but would be useful to be captured.

Redundant Information $I[X; Z \mid Y]$ quantifies information in the latent that is not needed for predicting the labels[5].

We also identify the following composite information quantities:

Relevant Information $I[X; Y] = I[X; Y \mid Z] + I[Y; Z]$ quantifies the information in the data that is relevant for the labels and which our model needs to capture to be able to predict the labels.

Preserved Information $I[X; Z] = I[X; Z \mid Y] + I[Y; Z]$ quantifies information from the data that is preserved in the latent.

Decoder Uncertainty $H[Y \mid Z] = I[X; Y \mid Z] + H[Y \mid X]$ quantifies the uncertainty about the labels after learning about the latent Z. If $H[Y \mid Z]$ reaches 0, it means that no additional information is needed to infer the correct label Y from the latent Z: the optimal decoder can be a deterministic mapping. Intuitively, we want to minimize this quantity for good predictive performance.

Reverse Decoder Uncertainty $H[Z \mid Y] = I[X; Z \mid Y] + H[Z \mid X]$ quantifies the uncertainty about the latent Z given the label Y. We can imagine training a new model to predict Z given Y and minimizing $H[Z \mid Y]$ to 0 would allow for a deterministic decoder from the latent to given the label.

Nuisance[6] $H[X \mid Y] = H[X \mid Y, Z] + I[X; Z]$ quantifies the information in the data that is not relevant for the task (Achille and Soatto, 2018a).

B.2 Definitions & equivalences

The following equalities can be read off from figure 2. For completeness and to provide a handy reference, we list them explicitly here. They can also be verified using symbolic manipulations and the properties of information quantities.

Equalities for composite quantities:

$$I[X; Y] = I[X; Y \mid Z] + I[Y; Z] \tag{17}$$
$$I[X; Z] = I[X; Z \mid Y] + I[Y; Z] \tag{18}$$
$$H[Y \mid Z] = I[X; Y \mid Z] + H[Y \mid X] \tag{19}$$
$$H[Z \mid Y] = I[X; Z \mid Y] + H[Z \mid X] \tag{20}$$
$$H[X \mid Y] = H[X \mid Y, Z] + I[X; Z] \tag{21}$$

We can combine the atomic quantities into the overall Label Entropy and Encoding Entropy:

$$H[Y] = H[Y \mid X] + I[Y; Z] + I[X; Y \mid Z] \tag{22}$$
$$H[Z] = H[Z \mid X] + I[Y; Z] + I[X; Z \mid Y]. \tag{23}$$

We can express the Relevant Information $I[X; Y]$, Residual Information $I[X; Y \mid Z]$, Redundant Information $I[X; Z \mid Y]$ and Preserved Information $I[X; Z]$ without X on the left-hand side:

$$I[X; Y] = H[Y] - H[Y \mid X], \tag{24}$$
$$I[X; Z] = H[Z] - H[Z \mid X], \tag{25}$$
$$I[X; Y \mid Z] = H[Y \mid Z] - H[Y \mid X], \tag{26}$$
$$I[X; Z \mid Y] = H[Z \mid Y] - H[Z \mid X]. \tag{27}$$

This simplifies estimating these expressions as X is usually much higher-dimensional and irregular than the labels or latent encodings. We also can rewrite the Preserved Relevant Information $I[Y; Z]$ as:

$$I[Y; Z] = H[Y] - H[Y \mid Z] \tag{28}$$
$$I[Y; Z] = H[Z] - H[Z \mid Y] \tag{29}$$

[5] Fisher (2019) uses the term "Residual Information" for this, which conflicts with Tishby and Zaslavsky (2015).

[6] Not depicted in figure 2.

C Information bottleneck & related works

C.1 Goals & motivation

The IB principle from Tishby et al. (2000) can be recast as a generalization of finding minimal sufficient statistics for the labels given the data (Shamir et al., 2010; Tishby and Zaslavsky, 2015; Fisher, 2019): it strives for minimality and sufficiency of the latent Z. Minimality is about minimizing amount of information necessary of X for the task, so minimizing the Preserved Information $I[X; Z]$; while sufficiency is about preserving the information to solve the task, so maximizing the Preserved Relevant Information $I[Y; Z]$.

From figure 2, we can read off the definitions of Relevant Information and Preserved Information:

$$I[X; Y] = I[Y; Z] + I[X; Y \mid Z] \tag{30}$$

$$I[X; Z] = I[Y; Z] + I[X; Z \mid Y], \tag{31}$$

and see that maximizing the Preserved Relevant Information $I[Y; Z]$ is equivalent to minimizing the Residual Information $I[X; Y \mid Z]$, while minimizing the Preserved Information $I[X; Z]$ at the same time means minimizing the Redundant Information $I[X; Z \mid Y]$, too, as $I[X; Y]$ is constant for the given dataset[7]. Moreover, we also see that the Preserved Relevant Information $I[Y; Z]$ is upper-bounded by Relevant Information $I[X; Y]$, so to capture all relevant information in our latent, we want $I[X; Y] = I[Y; Z]$.

Using the diagram, we can also see that minimizing the Residual Information is the same as minimizing the Decoder Uncertainty $H[Y \mid Z]$:

$$I[X; Y \mid Z] = H[Y \mid Z] - H[Y \mid X].$$

Ideally, we also want to minimize the Encoding Uncertainty $H[Z \mid X]$ to find the most deterministic latent encoding Z. Minimizing the Encoding Uncertainty and the Redundant Information $I[X; Z \mid Y]$ together is the same as minimizing the Reverse Decoder Uncertainty $H[Z \mid Y]$.

All in all, we want to minimize both the Decoder Uncertainty $H[Y \mid Z]$ and the Reverse Decoder Uncertainty $H[Z \mid Y]$.

C.2 IB objectives

"The Information Bottleneck Method" (IB)

Tishby et al. (2000) introduce $MI(X; \hat{X}) - \beta MI(\hat{X}; Y)$ as optimization objective for the Information Bottleneck. We can relate this to our notation by renaming $\hat{X} = Z$, such that the objective becomes "min $I[X; Z] - \beta I[Y; Z]$". The IB objective minimizes the Preserved Information $I[X; Z]$ and trades it off with maximizing the Preserved Relevant Information $I[Y; Z]$. Tishby and Zaslavsky (2015) mention that the IB objective is equivalent to minimizing $I[X; Z] + \beta I[X; Y \mid Z]$, see our discussion above. Tishby et al. (2000) provide an optimal algorithm for the tabular case, when X, Y and Z are all categorical. This has spawned additional research to optimize the objective for other cases and specifically for DNNs.

"Deterministic Information Bottleneck" (DIB)

Strouse and Schwab (2017) introduce as objective "min $H[Z] - \beta I[Y; Z]$". Compared to the IB objective, this also minimizes $H[Z \mid X]$ and encourages determinism. Vice-versa, for deterministic encoders, $H[Z \mid X] = 0$, and their objective matches the IB objective. Like Tishby et al. (2000), they provide an algorithm for the tabular case. To do so, they examine an analytical solution for their objective as it is unbounded: $H[Z \mid X] \to -\infty$ for the optimal solution. As we discuss in section 3.3, it does not easily translate to a continuous latent representation.

"Deep Variational Information Bottleneck"

Alemi et al. (2016) rewrite the terms in the bottleneck as maximization problem "max $I[Y; Z] - \beta I[X; Z]$" and swap the β parameter. Their β would be $1/\beta$ in IB above, which emphasizes that $I[Y; Z]$ is important for performance and $I[X; Z]$ acts as regularizer.

[7] That is, it does not depend on θ.

The paper derives the following variational approximation to the IB objective, where $z = f_\theta(x, \epsilon)$ denotes a stochastic latent embedding with distribution $\mathrm{p}_\theta(z \mid x)$, $\mathrm{p}_\theta(\hat{y} \mid z)$ denotes the decoder, and $\mathrm{r}(z)$ is some fixed prior distribution on the latent embedding:

$$\min \mathbb{E}_{\hat{\mathrm{p}}(x,y)} \mathbb{E}_{\epsilon \sim p(\epsilon)} \Big[-\log \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z = f_\theta(x_n, \epsilon)) + \gamma \, D_{\mathrm{KL}}(p(z|x_n) \| r(z)) \Big]. \tag{32}$$

In principle, the distributions $\mathrm{p}_\theta(\hat{y} \mid z)$ and $\mathrm{p}_\theta(z \mid x)$ could be given by arbitrary parameterizations and function approximators. In practice, the implementation of DVIB presented by Alemi et al. (2016) constructs $\mathrm{p}_\theta(z \mid x)$ as a multivariate Gaussian with parameterized mean and parameterized diagonal covariance using a neural network, and then uses a simple logistic regression to obtain $\mathrm{p}_\theta(\hat{y} \mid z)$, while arbitrarily setting $\mathrm{r}(z)$ to be a unit Gaussian around the origin. The requirement for $\mathrm{p}_\theta(z \mid x)$ to have a closed-form Kullback-Leibler divergence limits the applicability of the DVIB objective.

The DVIB objective can be written more concisely as

$$\min \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}] + \gamma \, D_{\mathrm{KL}}(\mathrm{p}(z \mid x) \,\|\, \mathrm{r}(z))$$

in the notation introduced in section 3. We discuss the regularizer in more detail in section F.3.

"Conditional Entropy Bottleneck"

Fisher (2019) introduce their Conditional Entropy Bottleneck as "$\min \mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}] - \mathrm{I}[\mathrm{Y}; \mathrm{Z}]$". We can rewrite the objective as $\mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}] + \mathrm{I}[\mathrm{X}; \mathrm{Y} \mid \mathrm{Z}] - \mathrm{I}[\mathrm{X}; \mathrm{Y}]$, using equations (30) and (31). The last term is constant for the dataset and can thus be dropped. Likewise, the IB objective can be rewritten as minimizing $\mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}] + (\beta - 1)\mathrm{I}[\mathrm{X}; \mathrm{Y} \mid \mathrm{Z}]$. The two match for $\beta = 2$. Fisher (2019) provides experimental results that favorably compare to Alemi et al. (2016), possibly due to additional flexibility as Fisher (2019) do not constrain $\mathrm{p}(z)$ to be a unit Gaussian and employ variational approximations for all terms.

C.3 Canonical IB & DIB objectives

We expand the IB and DIB objectives into "disjoint" terms and drop constant ones to find a more canonical form. This leads us to focus on the optimization of the Decoder Uncertainty $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ along with additional regularization terms. In section 3.2, we discuss the properties of $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$, and in section 3.3 we examine the regularization terms.

Proposition. *For IB, we obtain*

$$\arg\min \mathrm{I}[\mathrm{X}; \mathrm{Z}] - \beta \mathrm{I}[\mathrm{Y}; \mathrm{Z}] = \arg\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta' \underbrace{\mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}]}_{=\mathrm{H}[\mathrm{Z}|\mathrm{Y}] - \mathrm{H}[\mathrm{Z}|\mathrm{X}]}, \tag{33}$$

and, for DIB,

$$\arg\min \mathrm{H}[\mathrm{Z}] - \beta \mathrm{I}[\mathrm{Y}; \mathrm{Z}] = \arg\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta' \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] = \arg\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta'' \mathrm{H}[\mathrm{Z}] \tag{34}$$

with $\beta' := \frac{1}{\beta - 1} \in [0, \infty)$ *and* $\beta'' := \frac{1}{\beta} \in [0, 1)$.

Proof. For the steps marked with *, we make use of $\beta > 1$. For IB, we obtain

$$\begin{aligned} \arg\min \mathrm{I}[\mathrm{X}; \mathrm{Z}] - \beta \mathrm{I}[\mathrm{Y}; \mathrm{Z}] &= \arg\min \mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}] + (\beta - 1)\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] \\ &\overset{(*)}{=} \arg\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta' \, \mathrm{I}[\mathrm{X}; \mathrm{Z} \mid \mathrm{Y}] \\ &\arg\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta'(\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] - \mathrm{H}[\mathrm{Z} \mid \mathrm{X}]), \end{aligned} \tag{IB}$$

and, for DIB,

$$\begin{aligned} \arg\min \mathrm{H}[\mathrm{Z}] - \beta \mathrm{I}[\mathrm{Y}; \mathrm{Z}] &= \arg\min \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] + (\beta - 1)\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] \\ &\overset{(*)}{=} \arg\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta' \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}], \end{aligned} \tag{DIB}$$

with $\beta' := \frac{1}{\beta - 1} \in [0, \infty)$. Similarly, we show for DIB

$$\begin{aligned} \arg\min \mathrm{H}[\mathrm{Z}] - \beta \mathrm{I}[\mathrm{Y}; \mathrm{Z}] &= \arg\min \mathrm{H}[\mathrm{Z}] + \beta \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] \\ &\overset{(*)}{=} \arg\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta'' \mathrm{H}[\mathrm{Z}], \end{aligned}$$

with $\beta'' := \frac{1}{\beta} \in [0, 1)$, which is relevant in section 3.3.

We limit ourselves to $\beta > 1$, because, for $\beta < 1$, we would be maximizing the Decoder Uncertainty, which does not make sense: the obvious solution to this is one where Z contains no information

on Y, that is $p(y \mid z)$ is uniform. In the case of DIB, it is to map every input deterministically to a single latent; whereas for IB, we only minimize the Redundant Information, and the solution is free to contain noise. For $\beta = 1$, we would not care about Decoder Uncertainty and only minimize Redundant Information and Reverse Decoder Uncertainty, respectively, which allows for arbitrarily bad predictions. □

We note that we have $\beta' = \frac{\beta''}{1-\beta''}$ using the relations above.

C.4 IB objectives and the Entropy Distance Metric

Another perspective on the IB objectives is by expressing them using the Entropy Distance Metric. MacKay (2003, p. 140) introduces the entropy distance

$$EDM(\mathrm{Y}, \mathrm{Z}) = \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]. \tag{35}$$

as a metric when we identify random variables up to permutations of the labels for categorical variables: if the entropy distance is 0, Y and Z are the same distribution up to a consistent permutation of the labels (independent of X). If the entropy distance becomes 0, both $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] = 0 = \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$, and we can find a bijective map from Z to Y.[8]

We can express the Reverse Decoder Uncertainty $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$ using the Decoder Uncertainty $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ and the entropies:

$$\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] + \mathrm{H}[\mathrm{Y}] = \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \mathrm{H}[\mathrm{Z}],$$

and rewrite equation (35) as

$$EDM(\mathrm{Y}, \mathrm{Z}) = 2\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \mathrm{H}[\mathrm{Z}] - \mathrm{H}[\mathrm{Y}].$$

For optimization purposes, we can drop constant terms and rearrange:

$$\arg\min EDM(\mathrm{Y}, \mathrm{Z}) = \arg\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \tfrac{1}{2}\mathrm{H}[\mathrm{Z}].$$

C.4.1 Rewriting IB and DIB using the Entropy Distance Metric

For $\beta \geq 1$, we can rewrite equations (IB) and (DIB) as:

$$\arg\min EDM(\mathrm{Y}, \mathrm{Z}) + \gamma\,(\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] - \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]) + (\gamma - 1)\,\mathrm{H}[\mathrm{Z} \mid \mathrm{X}] \tag{36}$$

for IB, and

$$\arg\min EDM(\mathrm{Y}, \mathrm{Z}) + \gamma\,(\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] - \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]) \tag{37}$$

for DIB and replace β with $\gamma = 1 - \frac{2}{\beta} \in [-1, 1]$ which allows for a linear mix between $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ and $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$.

DIB will encourage the model to match both distributions for $\gamma = 0$ ($\beta = 2$), as we obtain a term that matches the Entropy Distance Metric from section C.4, and otherwise trades off Decoder Uncertainty and Reverse Decoder Uncertainty. IB behaves similarly but tends to maximize Encoding Uncertainty as $\gamma - 1 \in [-2, 0]$. Fisher (2019) argues for picking this configuration similar to the arguments in section C.1. DIB will force both distributions to become exactly the same, which would turn the decoder into a permutation matrix for categorical variables.

D Decoder Uncertainty $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$

D.1 Cross-entropy loss

The cross-entropy loss features prominently in section 3.2. We can derive the usual cross-entropy loss for our model by minimizing the Kullback-Leibler divergence between the empirical sample distribution $\hat{p}(x, y)$ and the parameterized distribution $p_\theta(x)\, p_\theta(\hat{y} \mid x)$. For discriminative models, we are only interested in $p_\theta(\hat{y} \mid x)$, and can simply set $p_\theta(x) = \hat{p}(x)$:

$$\arg\min_\theta D_{\mathrm{KL}}(\hat{p}(x, y) \parallel p_\theta(x)\, p_\theta(\hat{\mathrm{Y}} = y \mid x))$$

[8] The argument for continuous variables is the same. We need to identify distributions up to "isentropic" bijections.

$$= \arg\min_\theta D_{\mathrm{KL}}(\hat{\mathrm{p}}(y \mid x) \,\|\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid x)) + \underbrace{D_{\mathrm{KL}}(\hat{\mathrm{p}}(x) \,\|\, \mathrm{p}_\theta(x))}_{=0}$$

$$= \arg\min_\theta \mathrm{H}(\hat{\mathrm{p}}(y \mid x) \,\|\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid x)) - \underbrace{\mathrm{H}[\mathrm{Y} \mid \mathrm{X}]}_{\text{const.}}$$

$$= \arg\min_\theta \mathrm{H}(\hat{\mathrm{p}}(y \mid x) \,\|\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid x)).$$

In section 3.2, we introduce the shorthand $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}]$ for $\mathrm{H}(\hat{\mathrm{p}}(y \mid x) \,\|\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid x))$ and refer to it as Prediction Cross-Entropy.

D.2 Upper bounds & training error minimization

To motivate that $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ (or $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$) can be used as main loss term, we show that it can bound the (training) error probability since *accuracy* is often the true objective when machine learning models are deployed on real-world problems[9].

Proposition. *The Decoder Cross-Entropy provides an upper bound on the Decoder Uncertainty:*

$$\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] \leq \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \,\|\, \mathrm{p}_\theta(\hat{y} \mid z)) = \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}],$$

and further bounds the training error:

$$\mathrm{p}(\text{“}\hat{\mathrm{Y}} \textit{ is wrong”}) \leq 1 - e^{-\mathrm{H}_\theta[\mathrm{Y}|\mathrm{Z}]} = 1 - e^{-\left(\mathrm{H}[\mathrm{Y}|\mathrm{Z}] + D_{\mathrm{KL}}(\mathrm{p}(y|z) \| \mathrm{p}_\theta(\hat{y}|z))\right)}.$$

Likewise, for the Prediction Cross-Entropy $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}]$ and the Label Uncertainty $\mathrm{H}[\mathrm{Y} \mid \mathrm{X}]$.

Proof. The upper bounds for Decoder Uncertainty $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ and Label Uncertainty $\mathrm{H}[\mathrm{Y} \mid \mathrm{X}]$ follow from the non-negativity of the Kullback-Leibler divergence, for example:

$$0 \leq D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \,\|\, \mathrm{p}_\theta(\hat{y} \mid z)) = \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}] - \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}],$$

$$0 \leq D_{\mathrm{KL}}(\hat{\mathrm{p}}(y \mid x) \,\|\, \mathrm{p}_\theta(\hat{y} \mid x)) = \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}] - \mathrm{H}[\mathrm{Y} \mid \mathrm{X}].$$

The derivation for the training error probability is as follows:

$$\begin{aligned}\mathrm{p}(\text{“}\hat{\mathrm{Y}} \text{ is correct”}) &= \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, \mathrm{p}(\text{“}\hat{\mathrm{Y}} \text{ is correct”} \mid x, y) = \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, \mathbb{E}_{\mathrm{p}_\theta(z|x)}\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z) \\ &= \mathbb{E}_{\mathrm{p}(y,z)}\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z).\end{aligned}$$

We can then apply Jensen's inequality using convex $h(x) = -\ln x$:

$$\begin{aligned} & h\left(\mathbb{E}_{\mathrm{p}(y,z)}\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z)\right) \leq \mathbb{E}_{\mathrm{p}(y,z)}\, h\left(\mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z)\right) \\ \Leftrightarrow\ & \mathrm{p}(\text{“}\hat{\mathrm{Y}} \text{ is correct”}) \geq e^{-\mathrm{H}(\mathrm{p}(y|z) \| \mathrm{p}_\theta(\hat{\mathrm{Y}}=y|z))} \\ \Leftrightarrow\ & \mathrm{p}(\text{“}\hat{\mathrm{Y}} \text{ is wrong”}) \leq 1 - e^{-\mathrm{H}_\theta[\mathrm{Y}|\mathrm{Z}]}.\end{aligned}$$

For small $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$, we note that one can use the approximation $e^x \approx 1 + x$ to obtain:

$$\mathrm{p}(\text{“}\hat{\mathrm{Y}} \text{ is wrong”}) \lessapprox \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]. \tag{38}$$

Finally, we split the Decoder Cross-Entropy into the Decoder Uncertainty and a Kullback-Leibler divergence:

$$\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}] = \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \,\|\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z)).$$

If we upper-bound $D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \,\|\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z))$, minimizing the Decoder Uncertainty $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ becomes a sensible minimization objective as it reduces the probability of misclassification.

We can similarly show that the training error is bounded by the Prediction Cross-Entropy $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}]$. □

In the next section, we examine categorical Z for which optimal decoders can be constructed and $D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \,\|\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z))$ becomes zero.

[9] As we only take into account the empirical distribution $\hat{\mathrm{p}}(x, y)$ available for training, the following derivation refers only to the empirical risk, and not to the expected risk of the estimator $\hat{\mathrm{Y}}$.

E Categorical Z

For categorical Z, $\mathrm{p}(y \mid z)$ can be computed exactly for a given encoder $\mathrm{p}_\theta(z \mid x)$ by using the empirical data distribution, which, in turn, allows us to compute $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$[10]. This is similar to computing a confusion matrix between Y and Z but using information content instead of probabilities.

Moreover, if we set $\mathrm{p}_\theta(\hat{y} \mid z) := \mathrm{p}(\mathrm{Y} = \hat{y} \mid z)$ to have an optimal decoder, we obtain equality in equation (9), and obtain $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}] \leq \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}] = \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$. If the encoder were also deterministic, we would obtain $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}] = \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}] = \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$. We can minimize $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ directly using gradient descent. $\frac{d}{d\theta}\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ only depends on $\mathrm{p}(y \mid z)$ and $\frac{d}{d\theta}\,\mathrm{p}_\theta(z \mid x)$:

$$\frac{d}{d\theta}\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] = \mathbb{E}_{\mathrm{p}(x,z)}\left[\frac{d}{d\theta}\left[\ln \mathrm{p}_\theta(z \mid x)\right] \mathbb{E}_{\hat{\mathrm{p}}(y|x)}\, h\left(\mathrm{p}(y \mid z)\right)\right].$$

Proof.

$$\begin{aligned}
\frac{d}{d\theta}\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] &= \frac{d}{d\theta}\,\mathbb{E}_{\mathrm{p}(y,z)}\, h\left(\mathrm{p}(y \mid z)\right) = \frac{d}{d\theta}\,\mathbb{E}_{\mathrm{p}(x,y,z)}\, h\left(\mathrm{p}(y \mid z)\right) = \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\,\frac{d}{d\theta}\,\mathbb{E}_{\mathrm{p}_\theta(z|x)}\, h\left(\mathrm{p}(y \mid z)\right) \\
&= \mathbb{E}_{\mathrm{p}_\theta(z|x)}\,\mathbb{E}_{\hat{\mathrm{p}}(x,y)}\,\frac{d}{d\theta}\left[h\left(\mathrm{p}(y \mid z)\right)\right] + h\left(\mathrm{p}(y \mid z)\right)\frac{d}{d\theta}\left[\ln \mathrm{p}_\theta(z \mid x)\right] \\
&= \mathbb{E}_{\mathrm{p}(x,y,z)}\,\frac{d}{d\theta}\left[h\left(\mathrm{p}(y \mid z)\right)\right] + h\left(\mathrm{p}(y \mid z)\right)\frac{d}{d\theta}\left[\ln \mathrm{p}_\theta(z \mid x)\right].
\end{aligned}$$

And now we show that $\mathbb{E}_{\mathrm{p}(x,y,z)}\,\frac{d}{d\theta}\left[h\left(\mathrm{p}(y \mid z)\right)\right] = 0$:

$$\begin{aligned}
\mathbb{E}_{\mathrm{p}(x,y,z)}\,\frac{d}{d\theta}\left[h\left(\mathrm{p}(y \mid z)\right)\right] &= \mathbb{E}_{\mathrm{p}(y,z)}\,\frac{d}{d\theta}\left[h\left(\mathrm{p}(y \mid z)\right)\right] = \mathbb{E}_{\mathrm{p}(y,z)}\,\frac{-1}{\mathrm{p}(y \mid z)}\frac{d}{d\theta}\,\mathrm{p}(y \mid z) \\
&= -\int \frac{\mathrm{p}(y,z)}{\mathrm{p}(y \mid z)}\frac{d}{d\theta}\,\mathrm{p}(y \mid z)\,dy\,dz = -\int \mathrm{p}(z)\int \frac{d}{d\theta}\,\mathrm{p}(y \mid z)\,dy\,dz \\
&= -\int \mathrm{p}(z)\frac{d}{d\theta}\Big[\underbrace{\int \mathrm{p}(y \mid z)\,dy}_{=1}\Big]\,dz = 0.
\end{aligned}$$

Splitting the expectation and reordering of $\mathbb{E}_{\mathrm{p}(x,y,z)}\, h\left(\mathrm{p}(y \mid z)\right)\frac{d}{d\theta}\left[\ln \mathrm{p}_\theta(z \mid x)\right]$, we obtain the result. □

The same holds for Reverse Decoder Uncertainty $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$ and for the other quantities as can be verified easily.

If we minimize $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ directly, we can compute $\mathrm{p}(y \mid z)$ after every training epoch and fix $\mathrm{p}_\theta(\hat{y} \mid z) := \mathrm{p}(\mathrm{Y} = \hat{y} \mid z)$ to create the discriminative model $\mathrm{p}_\theta(\hat{y} \mid x)$. This is a different perspective on the self-consistent equations from Tishby et al. (2000); Gondek and Hofmann (2003).

E.1 Empirical evaluation of $D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \,\|\, \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z))$ during training

We examine the size of the gap between Decoder Uncertainty and Decoder Cross-Entropy and the training behavior of the two cross-entropies with *categorical* latent Z on Permutation MNIST and CIFAR-10. For Permutation MNIST (Goodfellow et al., 2013), we use the common fully-connected ReLU $784 - 1024 - 1024 - C$ encoder architecture, with $C = 100$ categories for Z. For CIFAR-10 (Krizhevsky et al., 2009), we use a standard ResNet18 model with C many output classes as encoder (He et al., 2016a). See section G for more details about the hyperparameters. Even though a $C \times 10$ matrix and a SoftMax would suffice to describe the decoder matrix $\mathrm{p}_\theta(\hat{y} \mid z)$[11], we have found that over-parameterization using a separate DNN benefits optimization a lot. Thus, to parameterize the decoder matrix, we use fully-connected ReLUs $C - 1024 - 1024 - 10$ with a final SoftMax layer. We compute it once per batch during training and back-propagate into it.

Figure E.1 shows the three metrics as we train with each of them in turn. Our results do not achieve SOTA accuracy on the test set—we impose a harder optimization problem as Z is categorical, and we

[10] $\mathrm{p}(y \mid z)$ depends on θ through $\mathrm{p}_\theta(z \mid x)$: $\mathrm{p}(y \mid z) = \frac{\sum_x \hat{\mathrm{p}}(x,y)\,\mathrm{p}_\theta(z|x)}{\sum_x \hat{\mathrm{p}}(x)\,\mathrm{p}_\theta(z|x)}$.

[11] For categorical Z, $\mathrm{p}_\theta(\hat{y} \mid z)$ is a stochastic matrix which sums to 1 along the $\hat{\mathrm{Y}}$ dimension.

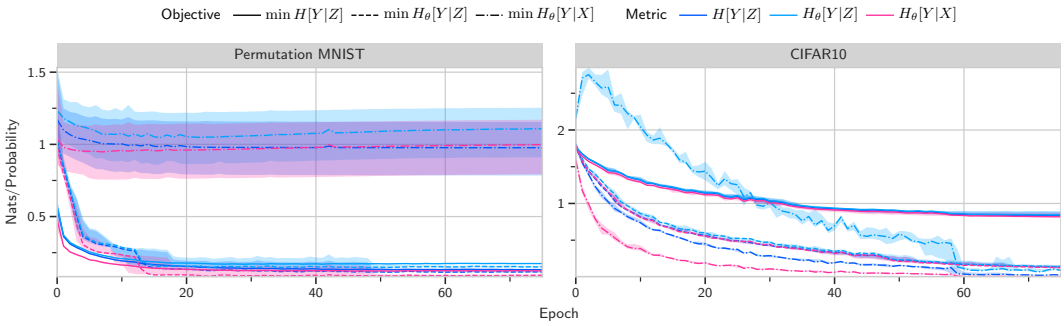


Figure E.1: *Decoder Uncertainty*, *Decoder Cross-Entropy* and *Prediction Cross-Entropy* for *Permutation-MNIST and CIFAR-10 with a categorical* Z. $C = 100$ categories are used for Z. We optimize with different minimization objectives in turn and plot the metrics. $D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \parallel \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z))$ is small when training with $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$ or $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$. When training with $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}]$ on CIFAR-10, $D_{\mathrm{KL}}(\mathrm{p}(y \mid z) \parallel \mathrm{p}_\theta(\hat{\mathrm{Y}} = y \mid z))$ remains quite large. We run 8 trials each and plot the median with confidence bounds (25% and 75% quartiles). See section E.1 for more details.

are essentially solving a hard-clustering problem first and then map these clusters to $\hat{\mathrm{Y}}$. Results are provided for the training set in order to compare with the optimal decoder.

As predicted, the Decoder Cross-Entropy upper-bounds both the Decoder Uncertainty $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ and the Prediction Cross-Entropy in all cases. Likewise, the gap between $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$ and $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$ is tiny when we minimize $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$. On the other hand, minimizing Prediction Cross-Entropy can lead to large gaps between $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$ and $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}]$, as can be seen for CIFAR-10.

Very interestingly, on MNIST Decoder Cross-Entropy provides a better training objective whereas on CIFAR-10 Prediction Cross-Entropy trains lower. Decoder Uncertainty does not train very well on CIFAR-10, and Prediction Cross-Entropy does not train well on Permutation MNIST at all. We suspect DNN architectures in the literature have evolved to train well with cross-entropies, but we are surprised by the heterogeneity of the results for the two datasets and models.

F Surrogates for Regularization Terms

F.1 Differential Entropies

Proposition. *After adding zero-entropy noise, the inequality* $\mathrm{I}[\mathrm{X};\mathrm{Z} \mid \mathrm{Y}] \leq \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] \leq \mathrm{H}[\mathrm{Z}]$ *also holds in the continuous case, and we can minimize* $\mathrm{I}[\mathrm{X};\mathrm{Z} \mid \mathrm{Y}]$ *in the IB objective by minimizing* $\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$ *or* $\mathrm{H}[\mathrm{Z}]$*, similarly to the DIB objective. We present a formal proof in section F.1.*

Theorem 2. *For random variables A, B, we have*

$$\mathrm{H}[A + B] \geq \mathrm{H}[B].$$

Proof. See Bercher and Vignat (2002, section 2.2). □

Proposition 1. *Let Y, Z and X be random variables satisfying the independence property* $Z \perp Y|X$*, and F a possibly stochastic function such that* $Z = F(X) + \epsilon$*, with independent noise* ϵ *satisfying* $\epsilon \perp F(X), \epsilon \perp Y$ *and* $H(\epsilon) = 0$*. Then the following holds whenever* $\mathrm{I}[\mathrm{Y};\mathrm{Z}]$ *is well-defined.*

$$\mathrm{I}[\mathrm{X};\mathrm{Z} \mid \mathrm{Y}] \leq \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] \leq \mathrm{H}[\mathrm{Z}].$$

Proof. First, we note that $\mathrm{H}[\mathrm{Z} \mid \mathrm{X}] = \mathrm{H}[F(X) + \epsilon \mid X] \geq \mathrm{H}[\epsilon \mid X] = \mathrm{H}[\epsilon]$ with theorem 2, as ϵ is independent of X, and thus $\mathrm{H}[\mathrm{Z} \mid \mathrm{X}] \geq 0$. We have $\mathrm{H}[\mathrm{Z} \mid \mathrm{X}] = \mathrm{H}[Z \mid X, Y]$ by the conditional independence assumption, and by the non-negativity of mutual information, $\mathrm{I}[\mathrm{Y};\mathrm{Z}] \geq 0$. Then:

$$\mathrm{I}[\mathrm{X};\mathrm{Z} \mid \mathrm{Y}] + \underbrace{\mathrm{H}[\mathrm{Z} \mid \mathrm{X}]}_{\geq 0} = \mathrm{H}[\mathrm{Z} \mid \mathrm{Y}]$$

$$\mathrm{H}[\mathrm{Z} \mid \mathrm{Y}] + \underbrace{\mathrm{I}[\mathrm{Y};\mathrm{Z}]}_{\geq 0} = \mathrm{H}[\mathrm{Z}]$$

□

The probabilistic model from section 2 fulfills the conditions exactly, and the two statements motivate our proposition.

It is important to note that while zero-entropy noise is necessary for preserving inequalities like $\mathrm{I}[\mathrm{X};\mathrm{Z}\mid\mathrm{Y}] \leq \mathrm{H}[\mathrm{Z}\mid\mathrm{Y}] \leq \mathrm{H}[\mathrm{Z}]$ in the continuous case, any Gaussian noise will suffice for optimization purposes: we optimize via pushing down an upper bound, and constant offsets will not affect this.

Thus, if we had $\mathrm{H}[\epsilon] \neq 0$, even though $\mathrm{I}[\mathrm{X};\mathrm{Z}\mid\mathrm{Y}] + \mathrm{H}[\mathrm{Z}\mid\mathrm{X}] \not\leq \mathrm{H}[\mathrm{Z}\mid\mathrm{Y}]$, we could instead use

$$\mathrm{I}[\mathrm{X};\mathrm{Z}\mid\mathrm{Y}] + \mathrm{H}[\mathrm{Z}\mid\mathrm{X}] - \mathrm{H}[\epsilon] \leq \mathrm{H}[\mathrm{Z}\mid\mathrm{Y}] - \mathrm{H}[\epsilon]$$

as upper bound to minimize. The gradients remain the same.

This also points to the nature of differential entropies as lacking a proper point of origin by themselves. We choose one by fixing $\mathrm{H}[\epsilon]$. Just like other literature usually only considers mutual information as meaningful, we consider $\mathrm{H}[\mathrm{Z}\mid\mathrm{X}] - \mathrm{H}[\epsilon]$ as more meaningful than $\mathrm{H}[\mathrm{Z}\mid\mathrm{X}]$. However, we can side-step this discussion conveniently by picking a canonical noise as point of origin in the form of zero-entropy noise $\mathrm{H}[\epsilon] = 0$.

F.2 Upper bounds

We derive this result as follows:

$$\begin{aligned}
\mathrm{H}[\mathrm{Z}\mid\mathrm{Y}] &= \mathbb{E}_{\hat{\mathrm{p}}(y)}\, \mathrm{H}[\mathrm{Z}\mid y] \\
&\leq \mathbb{E}_{\hat{\mathrm{p}}(y)}\, \tfrac{1}{2}\ln\det(2\pi e\ \mathrm{Cov}[\mathrm{Z}\mid y]) \\
&\leq \mathbb{E}_{\hat{\mathrm{p}}(y)} \sum_i \tfrac{1}{2}\ln(2\pi e\ \mathrm{Var}[\mathrm{Z}_i\mid y]) \\
&\approx \mathbb{E}_{\hat{\mathrm{p}}(y)} \sum_i \tfrac{1}{2}\ln(2\pi e\ \widehat{\mathrm{Var}}[\mathrm{Z}_i\mid y]),
\end{aligned}$$

Theorem 3. *Given a k-dimensional random variable $X = (X_i)_{i=1}^k$ with $\mathrm{Var}[X_i] > 0$ for all i,*

$$\begin{aligned}
\mathrm{H}[X] &\leq \tfrac{1}{2}\ln\det(2\pi e\ \mathrm{Cov}[X]) \\
&\leq \sum_i \tfrac{1}{2}\ln(2\pi e\ \mathrm{Var}[X_i]).
\end{aligned}$$

Proof. First, the multivariate normal distribution with same covariance is the maximum entropy distribution for that covariance, and thus $\mathrm{H}[X] \leq \ln\det(2\pi e\ \mathrm{Cov}[X])$, when we substitute the differential entropy for a multivariate normal distribution with covariance $\mathrm{Cov}[X]$. Let $\Sigma_0 := \mathrm{Cov}[X]$ be the covariance matrix and $\Sigma_1 := \mathrm{diag}(\mathrm{Var}[X_i])_i$ the matrix that only contains the diagonal. Because we add independent noise, $\mathrm{Var}[X_i] > 0$ and thus Σ_1^{-1} exists. It is clear that $\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) = k$. Then, we can use the KL-Divergence between two multivariate normal distributions $\mathcal{N}_0$, $\mathcal{N}_1$ with same mean 0 and covariances Σ_0 and Σ_1 to show that $\ln\det\Sigma_0 \leq \ln\det\Sigma_1$:

$$0 \leq D_{\mathrm{KL}}(\mathcal{N}_0 \parallel \mathcal{N}_1) = \tfrac{1}{2}\left(\mathrm{tr}(\Sigma_1^{-1}\Sigma_0) - k + \ln\left(\frac{\det\Sigma_1}{\det\Sigma_0}\right)\right)$$

$$\Leftrightarrow 0 \leq \tfrac{1}{2}\ln\left(\frac{\det\Sigma_1}{\det\Sigma_0}\right) \Leftrightarrow \tfrac{1}{2}\ln\det\Sigma_0 \leq \tfrac{1}{2}\ln\det\Sigma_1.$$

We substitute the definitions of Σ_0 and Σ_1, and obtain the second inequality after adding $k\ln(2\pi e)$ on both sides. □

Theorem 4. *Given a k-dimensional real-valued random variable $X = (X_i)_{i=1}^k \in \mathbb{R}^k$, we can bound the entropy by the mean squared norm of the latent:*

$$\mathbb{E}\,\|X\|^2 \leq C' \Rightarrow \mathrm{H}[X] \leq C, \tag{39}$$

with $C' := \frac{ke^{2C/k}}{2\pi e}$.

Proof. We begin with the previous bound:

$$\mathrm{H}[X] \leq \sum_i \tfrac{1}{2}\ln(2\pi e\ \mathrm{Var}[X_i]) = \tfrac{k}{2}\ln 2\pi e + \tfrac{1}{2}\ln\prod_i \mathrm{Var}[X_i]$$

$$\leq \frac{k}{2} \ln 2\pi e + \frac{1}{2} \ln \left(\frac{1}{k} \sum_i \mathrm{Var}[X_i] \right)^k = \frac{k}{2} \ln \frac{2\pi e}{k} \sum_i \mathrm{Var}[X_i]$$
$$\leq \frac{k}{2} \ln \frac{2\pi e}{k} \mathbb{E} \|X\|^2,$$

where we use the AM-GM inequality:

$$\left(\prod_i \mathrm{Var}[X_i] \right)^{\frac{1}{k}} \leq \frac{1}{k} \sum_i \mathrm{Var}[X_i]$$

and the monotony of the logarithm with:

$$\sum_i \mathrm{Var}[X_i] = \sum_i \mathbb{E}\left[X_i^2\right] - \mathbb{E}\left[X_i\right]^2 \leq \sum_i \mathbb{E}\left[X_i^2\right] = \mathbb{E}\|X\|^2$$

Bounding using $\mathbb{E}\|X\|^2 \leq C'$, we obtain

$$\mathrm{H}[X] \leq \frac{k}{2} \ln \frac{2\pi e}{k} C' = C,$$

and solving for C' yields the statement. □

This theorem provides justification for the use of $\ln \mathbb{E}\|Z\|^2$ as a regularizer, but does not justify the use of $\mathbb{E}\|Z\|^2$ directly. Here, we give two motivations. We first observe that $\ln x \leq x - 1$ due to ln's strict convexity and $\ln 1 = 0$, and thus:

$$\mathrm{H}[X] \leq \frac{k}{2} \ln \frac{2\pi e}{k} \mathbb{E}\|X\|^2 = \frac{k}{2} \left(\ln \frac{2\pi}{k} \mathbb{E}\|X\|^2 - 1 \right) \leq \pi \mathbb{E}\|X\|^2.$$

We can also take a step back and remind ourselves that IB objectives are actually Lagrangians, and β in $\min \mathrm{I}[\mathrm{X};\mathrm{Z}] - \beta \mathrm{I}[\mathrm{Y};\mathrm{Z}]$ is introduced as Lagrangian multiplier for the constrained objective:

$$\min \mathrm{I}[\mathrm{X};\mathrm{Z}] \text{ s.t. } \mathrm{I}[\mathrm{Y};\mathrm{Z}] \geq C.$$

We can similarly write our canonical DIB objective $\mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \beta'' \mathrm{H}[\mathrm{Z}]$ as constrained objective

$$\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] \text{ s.t. } \mathrm{H}[\mathrm{Z}] \leq C,$$

and use above statement to find the approximate form

$$\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] \text{ s.t. } \mathbb{E}\|Z\|^2 \leq C'.$$

Reintroducing a Lagrangian multiplier recovers our reguralized $\mathbb{E}\|Z\|^2$ objective:

$$\min \mathrm{H}[\mathrm{Y} \mid \mathrm{Z}] + \gamma \mathbb{E}\|Z\|^2.$$

F.3 Alemi et al. (2016) and $\mathbb{E}\|Z\|^2$

Alemi et al. (2016) model $\mathrm{p}_\theta(z \mid x)$ explicitly as multivariate Gaussian with parameterized mean and parameterized diagonal covariance in their encoder and regularize it to become close to $\mathcal{N}(0, I_k)$ by minimizing the Kullback-Leibler divergence $D_{\mathrm{KL}}(\mathrm{p}_\theta(z \mid x) \parallel \mathcal{N}(0, I_k))$ alongside the cross-entropy:

$$\min \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}] + \gamma\, D_{\mathrm{KL}}(\mathrm{p}(z \mid x) \parallel \mathrm{r}(z)),$$

as detailed in section C.2.

We can expand the regularization term to

$$\begin{aligned} & D_{\mathrm{KL}}(\mathrm{p}(z \mid x) \parallel \mathcal{N}(0, I_k)) \\ & \quad = \mathbb{E}_{\hat{\mathrm{p}}(x)} \mathbb{E}_{\mathrm{p}(z|x)}\, h\left((2\pi)^{-\frac{k}{2}} e^{-\frac{1}{2}\|\mathrm{Z}\|^2 |} \right) - \mathrm{H}[\mathrm{Z} \mid \mathrm{X}] \\ & \quad = \mathbb{E}_{\mathrm{p}(z)} \left[\frac{k}{2} \ln(2\pi) + \frac{1}{2}\|\mathrm{Z}\|^2 \right] - \mathrm{H}[\mathrm{Z} \mid \mathrm{X}]. \end{aligned}$$

After dropping constant terms (as they don't matter for optimization purposes), we obtain

$$= \frac{1}{2} \mathbb{E}\|Z\|^2 - \mathrm{H}[\mathrm{Z} \mid \mathrm{X}].$$

When we inject zero-entropy noise into the latent Z, we have $\mathrm{H}[\mathrm{Z} \mid \mathrm{X}] \geq 0$ and thus $\mathbb{E}\|Z\|^2 - \mathrm{H}[\mathrm{Z} \mid \mathrm{X}] \leq \mathbb{E}\|Z\|^2$. Thus, the $\mathbb{E}\|Z\|^2$ regularizer also upper-bounds DVIB's regularizer in this case.

In particular, we have equality when we use a deterministic encoder. When we inject zero-entropy noise and use a deterministic encoder, we are optimizing the DVIB objective function when we use the $\mathbb{E}\|Z\|^2$ regularizer. In other words, in this particular case, we could reinterpret "$\min \mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}] +$

$\gamma\, \mathbb{E}\, \|Z\|^2$" as optimzing the DVIB objective from Alemi et al. (2016) if they were using a constant covariance instead of parameterizing it in their encoder. This does not hold for stochastic encoders.

F.4 Soft clustering by entropy Minimization with Gaussian noise

Consider the problem of minimizing $\mathrm{H}[Z \mid Y]$ and $\mathrm{H}[Y \mid Z]$, in the setting where $Z = f_\theta(X) + \epsilon \sim \mathcal{N}(0, \sigma^2)$—i.e. the embedding Z is obtained by adding Gaussian noise to a deterministic function of the input. Let the training set be enumerated $x_1, \ldots, x_n$, with $\mu_i = f_\theta(x_i)$. Then the distribution of Z is given by a mixture of Gaussians with the following density, where $d(x, \mu_i) := \|x - \mu_i\|/\sigma^2$.

$$\mathrm{p}(z) \propto \frac{1}{n} \sum_{i=1}^{n} \exp(-d(z, \mu_i))$$

Assuming that each x_i has a deterministic label y_i, we then find that the conditional distributions $\mathrm{p}(y \mid z)$ and $\mathrm{p}(z \mid y)$ are given as follows:

$$\begin{aligned}
\mathrm{p}(z \mid y) &\propto \frac{1}{n_y} \sum_{i:y_i=y} \exp(-d(z, \mu_i)) \\
\mathrm{p}(y \mid z) &= \sum_{i:y_i=y} \mathrm{p}(\mu_i \mid z) = \sum_{i:y_i=y} \frac{\mathrm{p}(z \mid \mu_i)\, \mathrm{p}(\mu_i)}{\mathrm{p}(z)} \\
&= \frac{\sum_{i:y_i=y} \mathrm{p}(z \mid \mu_i)}{\sum_{k=1}^{n} \mathrm{p}(z \mid \mu_k)} = \frac{\sum_{i:y_i=y} \exp(-d(z, \mu_i))}{\sum_{k=1}^{n} \exp(-d(z, \mu_k))},
\end{aligned}$$

where n_y is the number of x_i with class $y_i = y$. Thus, the conditional $Z|Y$ can be interpreted as a mixture of Gaussians and $Y|Z$ as a Softmax marginal with respect to the distances between Z and the mean embeddings. We observe that $\mathrm{H}[Z \mid Y]$ is lower-bounded by the entropy of the random noise added to the embeddings:

$$\mathrm{H}[Z \mid Y] \geq \mathrm{H}[f_\theta(X) + \epsilon \mid Y] \geq \mathrm{H}[\epsilon]$$

with equality when the distribution of $f_\theta(X)|Y$ is deterministic – that is f_θ is constant for each equivalence class.

Further, the entropy $\mathrm{H}[Y \mid Z]$ is minimized when $\mathrm{H}[Z]$ is large compared to $\mathrm{H}[Z \mid Y]$ as we have the decomposition

$$\mathrm{H}[Y \mid Z] = \mathrm{H}[Z \mid Y] - \mathrm{H}[Z] + \mathrm{H}[Y].$$

In particular, when f_θ is constant over equivalence classes of the input, then $\mathrm{H}[Y \mid Z]$ is minimized when the entropy $H[f_\theta(X) + \epsilon]$ is large – i.e. the values of $f_\theta(x_i)$ for each equivalence class are distant from each other and there is minimal overlap between the clusters. Therefore, the optima of the information bottleneck objective under Gaussian noise share similar properties to the optima of geometric clustering of the inputs according to their output class.

To gain a better understanding of local optimization behavior, we decompose the objective terms as follows:

$$\begin{aligned}
\mathrm{H}[Z \mid Y] &= \mathbb{E}_{\hat{\mathrm{p}}(y)}\, \mathrm{H}(\mathrm{p}(z \mid y) \,\|\, \mathrm{p}(z \mid y)) \\
&= \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, \mathrm{H}(\mathrm{p}(z \mid x) \,\|\, \mathrm{p}(z \mid y)) \\
&= \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, D_{\mathrm{KL}}(\mathrm{p}(z \mid x) \,\|\, \mathrm{p}(z \mid y)) + \mathrm{H}[Z \mid x] \\
&= \mathbb{E}_{\hat{\mathrm{p}}(x,y)}\, D_{\mathrm{KL}}(\mathrm{p}(z \mid x) \,\|\, \mathrm{p}(z \mid y)) \\
&\quad + \underbrace{\mathrm{H}[Z \mid X]}_{=const}.
\end{aligned}$$

To examine how the mean embedding μ_k of a single datapoint x_k affects this entropy term, we look at the derivative of this expression with respect to $\mu_k = f_\theta(x_k)$. We obtain:

$$\begin{aligned}
\frac{d}{d\mu_k} \mathrm{H}[Z \mid Y] &= \frac{d}{d\mu_k}\, \mathrm{H}[Z \mid y_k] \\
&= \frac{d}{d\mu_k}\, \mathbb{E}_{\mathrm{p}(x|y_k)}\, D_{\mathrm{KL}}(\mathrm{p}(z \mid x) \,\|\, \mathrm{p}(z \mid y)) \\
&= \sum_{i \neq i:y_i=y_k} \frac{1}{n_{y_k}} \frac{d}{d\mu_k} D_{\mathrm{KL}}(\mathrm{p}(z \mid x_i) \,\|\, \mathrm{p}(z \mid y_k))
\end{aligned}$$

$$+ \frac{1}{n_{y_k}} \frac{d}{d\mu_k} D_{\mathrm{KL}}(\mathrm{p}(z \mid x_k) \,\|\, \mathrm{p}(z \mid y_k)).$$

While these derivatives do not have a simple analytic form, we can use known properties of the KL divergence to develop an intuition on how the gradient will behave. We observe that in the left-hand sum μ_k only affects the distribution of $Z|Y$ (that is we are differentiating a sum of terms that look like a reverse KL), whereas it has greater influence on $\mathrm{p}(z \mid x_k)$ in the right-hand term, and so its gradient will more closely resemble that of the forward KL. The left-hand-side term will therefore push μ_k towards the centroid of the means of inputs mapping to y, whereas the right-hand side term is mode-seeking.

F.5 A note on differential and discrete entropies

The mutual information between two random variables can be defined in terms of the KL divergence between the product of their marginals and their joint distribution. However, the KL divergence is only well-defined when the Radon-Nikodym derivative of the density of the joint with respect to the product exists. Mixing continuous and discrete distributions—and thus differential and continuous entropies—can violate this requirement, and so lead to negative values of the "mutual information". This is particularly worrying in the setting of training stochastic neural networks, as we often assume that an stochastic embedding is generated as a deterministic transformation of an input from a finite dataset to which a continuous perturbation is added. We provide an examples where naive computation without ensuring that the product and joint distributions of the two random variables have a well-defined Radon-Nikodym derivative yields negative mutual information.

Let $X \sim U([0, 0.1])$, $Z = X + R$ with $R \sim U(\{0, 1\})$. Then

$$\mathrm{I}[X; Z] = \mathrm{H}[X] = \log \tfrac{1}{10} \leq 0.$$

Generally, given X as above and an invertible function f such that $Z = f(X)$, $\mathrm{I}[X; Z] = \mathrm{H}[X]$ and can thus be negative. In a way, these cases can be reduced to (degenerate) expressions of the form $\mathrm{I}[X; X] = \mathrm{H}[X]$.

We can avoid these cases by adding independent continuous noise.

These examples show that not adding noise can lead to unexpected results. While they still yield finite quantities that bear a relation to the entropies of the random variables, they violate some of the core assumptions we have such that mutual information is always positive.

G Experiment details

G.1 DNN architectures and hyperparameters

For our experiments, we use PyTorch (Paszke et al., 2019) and the Adam optimizer (Kingma and Ba, 2014). In general, we use an initial learning rate of 0.5×10^{-3} and multiply the learning rate by $\sqrt{0.1}$ whenever the loss plateaus for more than 10 epochs for CIFAR-10. For MNIST and Permutation MNIST, we use an initial learning rate of 10^{-4} and multiply the learning rate by 0.8 whenever the loss plateaus for more than 3 epochs.

Sadly, we deviate from this in the following experiments: when optimizing the decoder uncertainty for *categorical* Z for CIFAR-10, we used 5 epochs patience for the decoder uncertainty objective and a initial learning rate of 10^{-4}. We do not expect this difference to affect the qualitative results mentioned in section E when comparing to other objectives. We also only used 5 epochs patience when comparing the two cross-entropies on CIFAR-10 in section 3.2. As this was used for both sets of experiments, it does not matter.

We train the experiments for creating the information plane plots for 150 epochs. The toy experiment (figure 3) is trained for 20 epochs. All other experiments train for 100 epochs.

We use a batchsize of 128 for most experiments. We use a batchsize of 32 for comparing the cross-entropies for CIFAR-10 (where we take 8 Dropout samples each), and a batchsize of 16 for MNIST (where we take 64 Dropout samples each).

For MNIST, we use a standard Dropout CNN, following `https://github.com/pytorch/examples/blob/master/mnist/main.py`. For Permutation MNIST, we use a fully-connected model (for experiments with categorical Z in section E): $784 \times 1024 \times 1024 \times C$. For CIFAR-10, we use a regular deterministic ResNet18 model (He et al., 2016a) for the experiments in section E. (As the model outputs a categorical distribution it becomes stochastic through that and we don't need stochasticity in the weights.) For the other experiments as well as the ImageNette experiments, we use a ResNet18v2 (He et al., 2016b). When we need a stochastic model for CIFAR-10 (for *continuous* Z), we add DropConnect (Wan et al., 2013b) with rate 0.1 to all but the first convolutional layers and Dropout with rate 0.1 before the final fully-connected layer. Because of memory issues, we reuse the Dropout masks within one batch. The model trains to 94% accuracy on CIFAR-10.

For CIFAR-10, we always remove the maximum pooling layer and change the first convolutional layer to have kernel size 3 with stride 1 and padding 1. We also use dataset augmentation during training, but not during evaluation on the training set and test set for purposes of computing metrics. We crop randomly after the padding the training images by 4 pixels in every direction and randomly flip images horizontally.

We generally sample 30 values of γ for the information plane plots from the specified ranges, using a log scale. For the ablation studies mentioned below, we sample 10 values of γ each. We always sample $\gamma = 0$ separately and run a trial with it.

Baselines were tuned by hand (without regularization) using grad-student descent and small grid searches.

G.2 Cluster setup & used resources

We make use of a local SLURM cluster (Jette et al., 2002). We run our experiments on GPUs (Geforce RTX 2080 Ti). We estimate reproducing all results would take 94 GPU days.

G.3 Comparison of the surrogate objectives

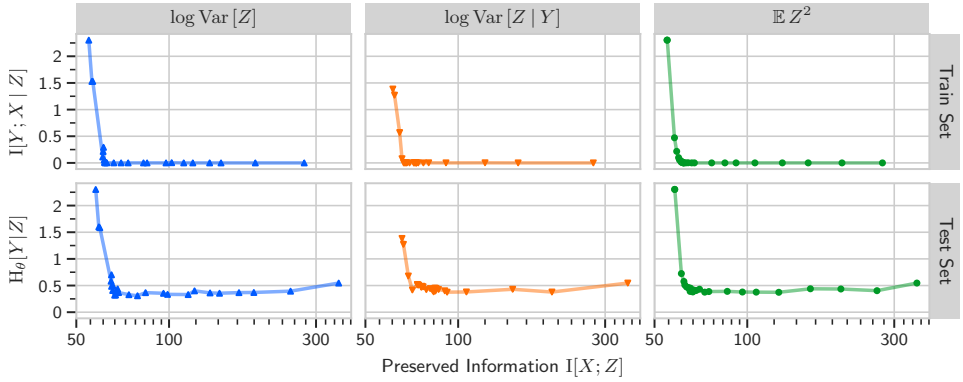


Figure G.1: *Information Plane Plot of the latent* Z *similar to Tishby and Zaslavsky (2015) but using a* ResNet18 *model on* CIFAR-10 *using the different regularizes from section 3.3 (*without Dropout, but with zero-entropy noise*).* The dots are colored by γ. See section 4 for more details.

As can be seen in figure G.2, the different surrogate regularizers have very similar effects on H[Z] and H[Z | Y]. Regularizing with $\mathbb{E}\,\|Z\|^2$ shows a stronger initial regularization effect, but is difficult to compare quantitatively as its hyperparameter does not map to an equivalent β, unlike regularizing using entropy estimates. Overall, we find log Var[Z] to provide stable training trajectories (and expected visualizations) while also having a more meaningful hyperparameter than $\mathbb{E}\,\|Z\|^2$, though $\mathbb{E}\,\|Z\|^2$ is trivial to implement and communicate[12]. log Var[Z | Y] performs worse which we hypothesize is due to the increased variance (given equal batch sizes) from conditioning on Y. It further does not minimize the Preserved Information I[X; Z] as strongly as the other regularizers.

[12]Which is the reason why we showcase it in figure 1 and in equation (1).

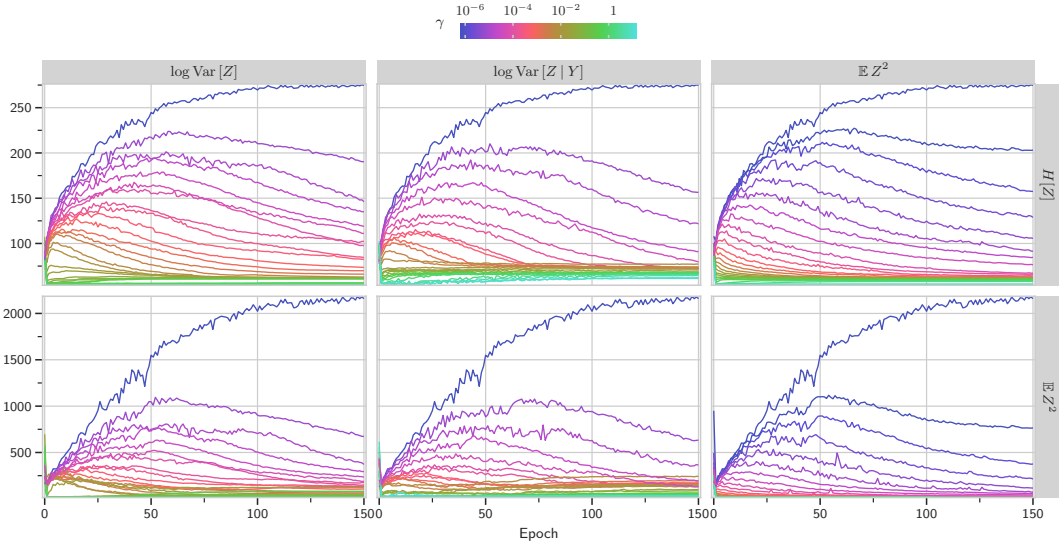


Figure G.2: *Entropy estimates while training with different γ and with different surrogate regularizers on* CIFAR-10 *with a* ResNet18 *model.* Entropies are estimated on training data based on Kraskov et al. (2004). Qualitatively all three regularizers push H[Z] and H[Z | Y] down. H[Z | Y] is not shown here because it always stays very close to H[Z]. $\mathbb{E}\,\|Z\|^2$ tends to regularize entropies more strongly for small γ. See section 4 for more details.

G.3.1 Measurements of information quantities

Measuring information quantities can be challenging. As mentionend in the introduction, there are many complex ways of measuring entropies and mutual information terms. We can side-step the issue by making use of the bounds we have established and the zero-entropy noise we are injecting, and design experiments around that.

First, to estimate the Preserved Information I[X; Z], we note that when we use a deterministic model as encoder and only inject zero-entropy noise, we have H[Z | X] = 0 and I[X; Z] = I[X; Z] + H[Z | X] = H[Z]. We use the entropy estimator from Kraskov et al. (2004, equation (20)) to estimate the Encoding Entropy H[Z] and thus I[X; Z].

To estimate the Residual Information I[X; Y | Z], we similarly note that I[X; Y | Z] = I[X; Y | Z] + H[Y | X] = H[Y | Z]. Instead of estimating the entropy using Kraskov et al. (2004), we can use the Decoder Cross-Entropy H_θ[Y | Z] which provides a tighter bound as long as we also minimize H_θ[Y | Z] as part of the training objective.

When we use stochastic models as encoder, we cannot easily compute I[X; Z] anymore. In the ablation study in the next section, we thus change the X axis accordingly.

Similarly, when we look at the trajectories on the test set instead of the training set, for example in figure G.4, we change the Y axis to signify the Decoder Uncertainty H_θ[Y | Z]. It is still an upper-bound, but we do not minimize it directly anymore.

For the plots in figure 6, we retrained the decoder on the test set to obtain a tighter bound on H[Y | Z] (while keeping the encoder fixed). We then sampled the latent using the test set to estimate the trajectories. We only did this for the CIFAR-10 model without dropout. For our ablations, we did not retrain the decoder and thus only present plots on the test and training set, respectively.

At this point, it is important to recall that the Decoder Uncertainty is also the negative log-likelihood (when training with a single Dropout sample), which provides a different perspective on the plots. It makes it clear that we can see how much a model overfits by comparing the best and final epochs of a trajectory in the plot (marked by a circle and a square, respectively).

G.3.2 Ablation study

We perform an ablation study to determine whether injecting noise is necessary. Furthermore, we investigate the more interesting case of using a stochastic model as encoder, and if we can use a stochastic model without injecting zero-entropy noise.

We also investigate whether $\log \mathrm{Var}[Z \mid Y]$ performs better when we increase batchsize as we hypothesized that a batchsize of 128 does not suffice as it leaves only ≈ 13 samples per class to approximate $\mathrm{H}[Z \mid Y]$).

Figure G.3 shows a larger version of figure 6 for all three regularizers and also training trajectories on the test set. As described in the previous section, this allows us to validate that the regularizers prevent overfitting on the training set: with increasing γ, the model overfits less.

Figure G.6 and figure G.5 shows that injecting noise is necessary independently of whether we use Dropout or not. Regularizing with $\mathbb{E}\,\|Z\|^2$ still has a very weak effect. We hypothesize that, similar to the toy experiment depicted in figure 3, floating-point precision issues might provide a natural noise source eventually. This would change the effectiveness of γ and might require much higher values to observe similar regularization effects as when we do inject zero-entropy noise.

Figure G.4 shows trajectories for a stochastic encoder (as described above with DropConnect/Dropout rate 0.1). It overfits less than a deterministic one.

Figure G.7 shows the effects of using higher dropout rates (using DropConnect/Dropout rates of 0.3/0.5). It overfits less than model with DropConnect/Dropout rates of 0.1/0.1.

The plots in figure G.10 show the effects of different γ with different regularizers more clearly. On both training and test set, one can clearly see the effects of regularization.

Overall, $\log \mathrm{Var}[Z \mid Y]$ performs worse as a regularizer. In figure G.8, we compare the effect of doubling batchsize. Indeed, $\log \mathrm{Var}[Z \mid Y]$ performs better with higher batchsize and looks closer to $\log \mathrm{Var}[Z]$.

G.3.3 Comparison between Decoder Cross-Entropy and Prediction Cross-Entropy

When training deterministic models or Dropout models with a single sample (as one usually does), the estimators for both the Decoder Cross-Entropy $\mathrm{H}_\theta[Y \mid Z]$ and the Prediction Cross-Entropy $\mathrm{H}_\theta[Y \mid X]$ coincide. In section 3.2, we discuss the differences from a theoretical perspective. Here, we empirically evaluate the difference between optimizing the estimators for each of the two cross-entropy losses, for which we will draw multiple Dropout samples during training and inference.

We examine models with *continuous* Z on MNIST and CIFAR-10 (Lecun et al., 1998; Krizhevsky et al., 2009). Specifically, we use a standard Dropout CNN as an encoder for MNIST, and a modified ResNet18 to which we add DropConnect in each layer for CIFAR-10. We use $K = 100$ dimensions for the continuous latent Z in the last fully-connected layer, and use a linear decoder to obtain the final 10-dimensional output of class logits. For MNIST, we compute the cross-entropies using 64 Dropout samples; for CIFAR-10, we use 8. For the purpose of this examination of training behavior, it is not necessary to achieve SOTA accuracy: our models obtain 99.2% accuracy on MNIST and 93.6% on CIFAR-10.

Figure G.11 shows the training error probability as well as the value of each cross-entropy loss for models trained either with the Decoder Cross-Entropy or the Prediction Cross-Entropy. The Decoder Cross-Entropy $\mathrm{H}_\theta[Y \mid Z]$ outperforms Prediction Cross-Entropy $\mathrm{H}_\theta[Y \mid X]$ as a training objective: the training error probability and both cross-entropies are lower when minimizing $\mathrm{H}_\theta[Y \mid Z]$ compared to minimizing $\mathrm{H}_\theta[Y \mid X]$. We compare only the training, rather than the test, losses of the models to isolate the effect of each loss term on training performance; we leave the prevention of overfitting to the regularization terms considered later. Recently, Dusenberry et al. (2020) also observed empirically that the Decoder Cross-Entropy $\mathrm{H}_\theta[Y \mid Z]$ as an objective is both easier to optimize and provides better generalization performance.

G.4 Differential entropies and noise

We demonstrate the importance of adding noise to continuous latents by constructing a pathological sequence of parameters which attain monotonically improving and unbounded regularized objective values (H[Z]) while all computing *the same function*. We use MNIST with a standard Dropout CNN as encoder, with $K = 128$ continuous dimensions in Z, and a $K \times 10$ linear layer as decoder. After every training epoch, we decrease the entropy of the latent by normalizing and then scaling the latent to bound the entropy. We multiply the weights of the decoder to not change the overall function. As can be seen in figure 3, without noise, entropy can decrease freely during training without change in error rate until it is affected by floating-point issues; while when adding zero-entropy noise, the error rate starts increasing gradually and meaningfully as the entropy starts to approach zero. We conclude that entropy regularization is meaningful only when noise is added to the latent.

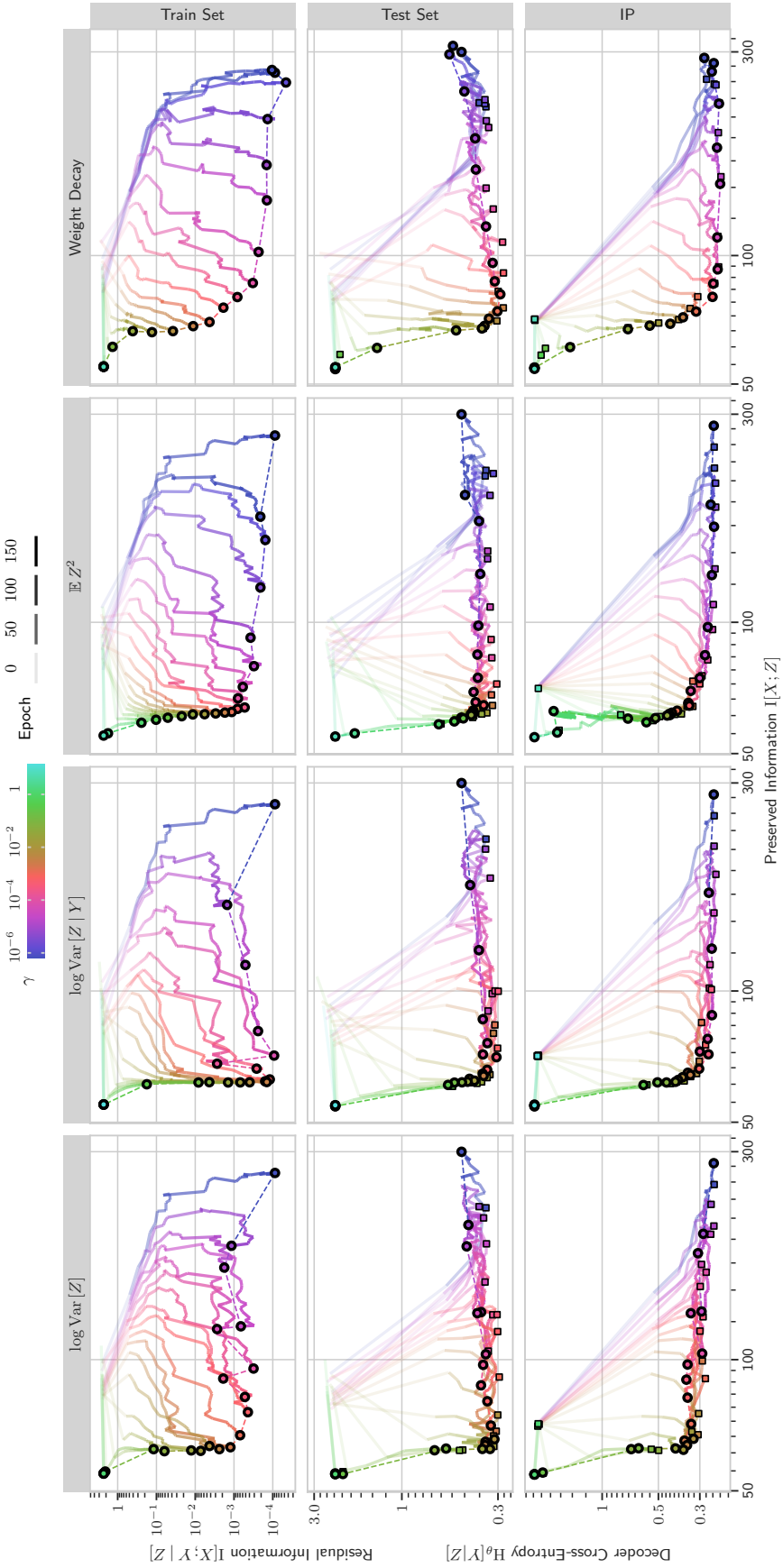


Figure G.3: Without dropout but with zero-entropy noise: *Information Plane Plot of training trajectories for ResNet18 models on CIFAR-10 and different regularizers.* The trajectories are colored by their respective γ; their transparency changes by epoch. Compression (Preserved Information ↓) trades-off with performance (Residual Information ↓). See section 4. The circle marks the final epoch of a trajectory. The square marks the best epoch (Residual Information ↓↓).

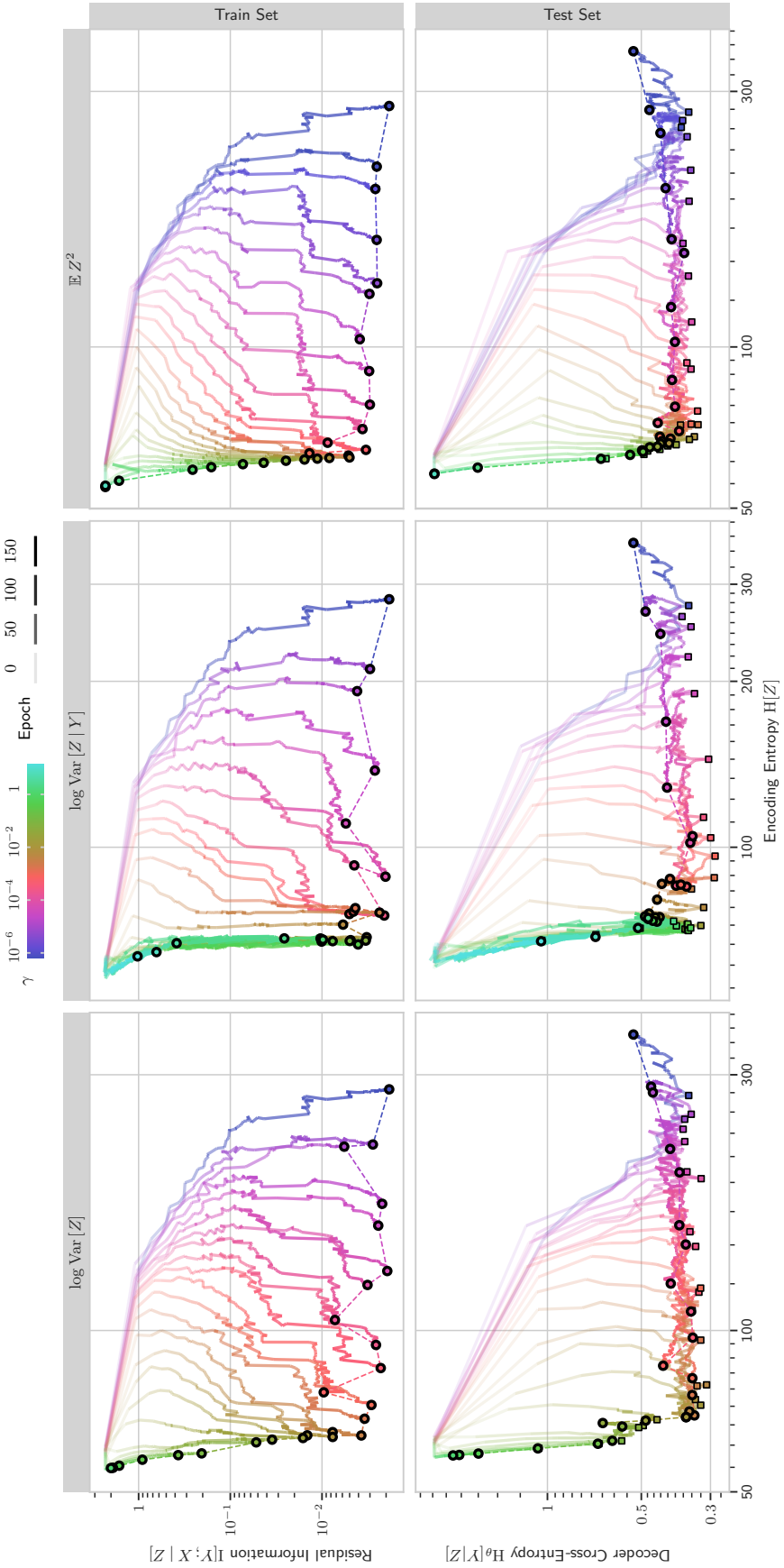


Figure G.4: With dropout and with zero-entropy noise: *Information Plane Plot of training trajectories for ResNet18 models on CIFAR-10 and different regularizers.* The trajectories are colored by their respective γ; their transparency changes by epoch. Compression (Preserved Information ↓) trades-off with performance (Residual Information ↓). See section 4. The circle marks the final epoch of a trajectory. The square marks the best epoch (Residual Information ↓↓).

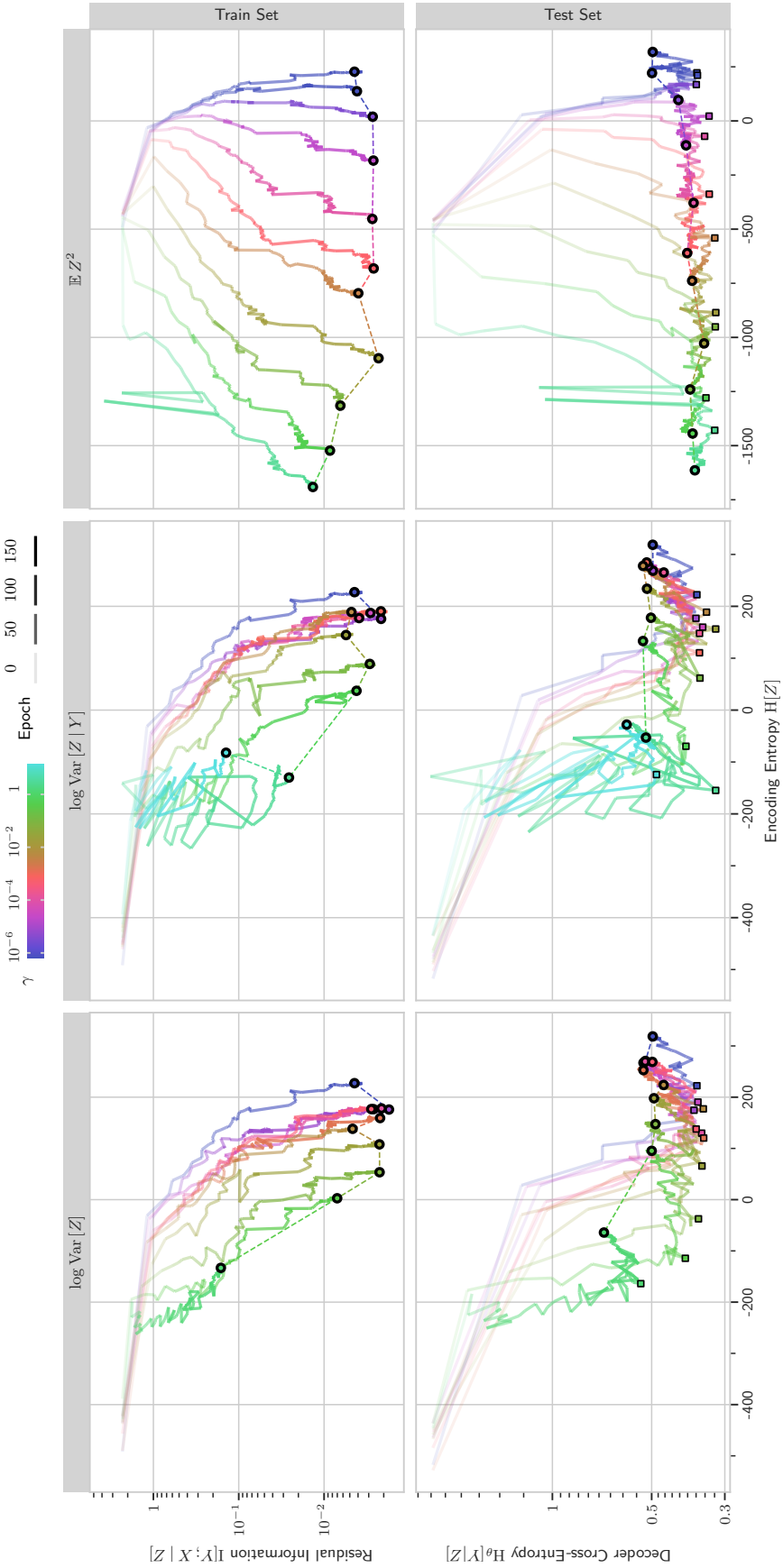


Figure G.5: With dropout but without zero-entropy noise: *Information Plane Plot of training trajectories for ResNet18 models on CIFAR-10 and different regularizers.* The trajectories are colored by their respective γ; their transparency changes by epoch. Compression (Preserved Information ↓) trades-off with performance (Residual Information ↓). See section 4. The circle marks the final epoch of a trajectory. The square marks the best epoch (Residual Information ↓↓).

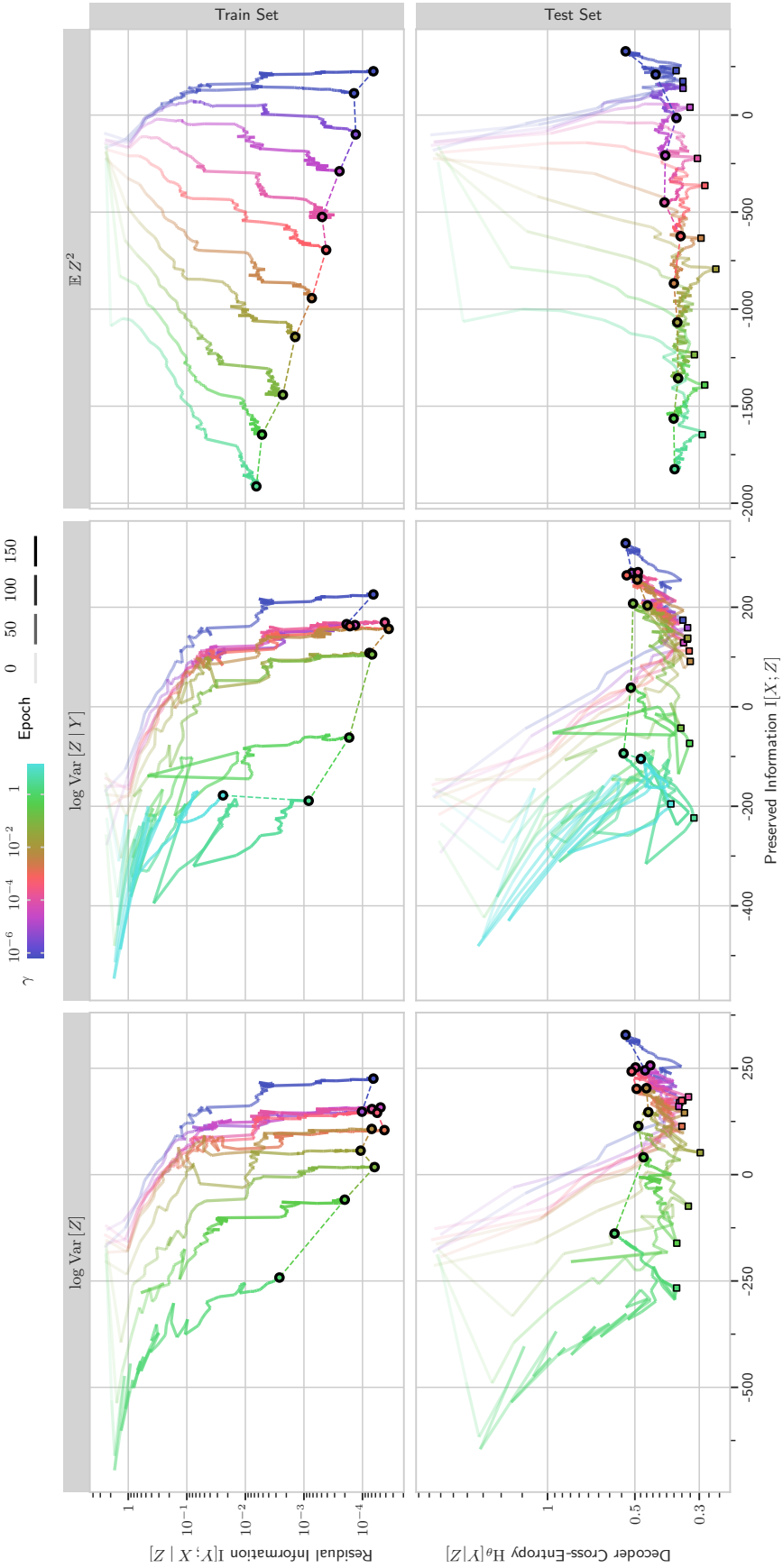


Figure G.6: Without dropout and without zero-entropy noise: *Information Plane Plot of training trajectories for ResNet18 models on CIFAR-10 and different regularizers*. The trajectories are colored by their respective γ; their transparency changes by epoch. Compression (Preserved Information ↓) trades-off with performance (Residual Information ↓). See section 4. The circle marks the final epoch of a trajectory. The square marks the best epoch (Residual Information ↓↓).

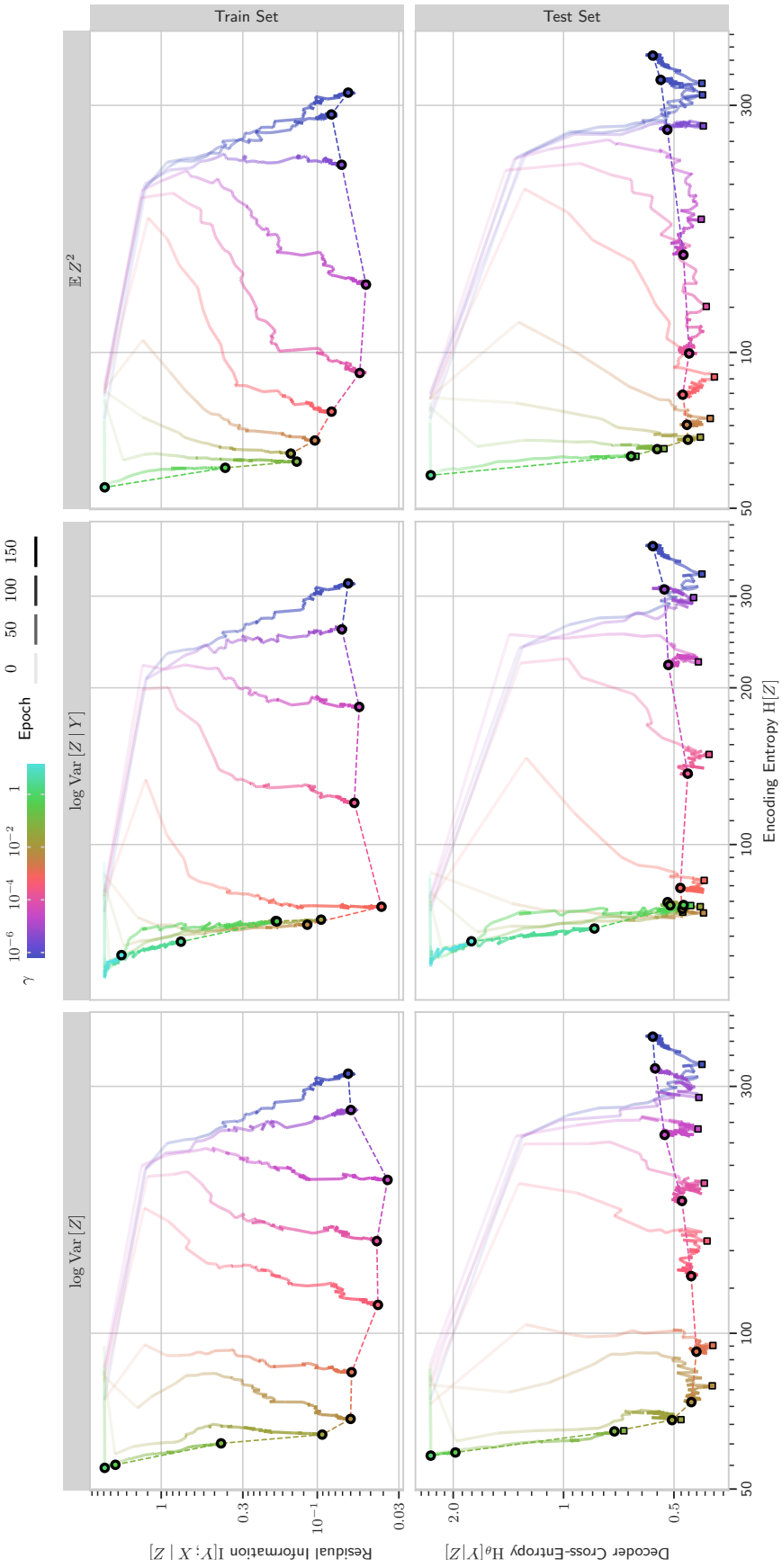


Figure G.7: With more dropout and zero-entropy noise: *Information Plane Plot of training trajectories for ResNet18 models on CIFAR-10 and* log Var[Z | Y] *regularizer with batchsizes 128 and 256.* The trajectories are colored by their respective γ; their transparency changes by epoch. Compression (Preserved Information ↓) trades-off with performance (Residual Information ↓). See section 4. The circle marks the final epoch of a trajectory. The square marks the best epoch (Residual Information ↓↓). A DropConnect rate of 0.3 and Dropout rate of 0.4 were used instead of 0.1 for each.

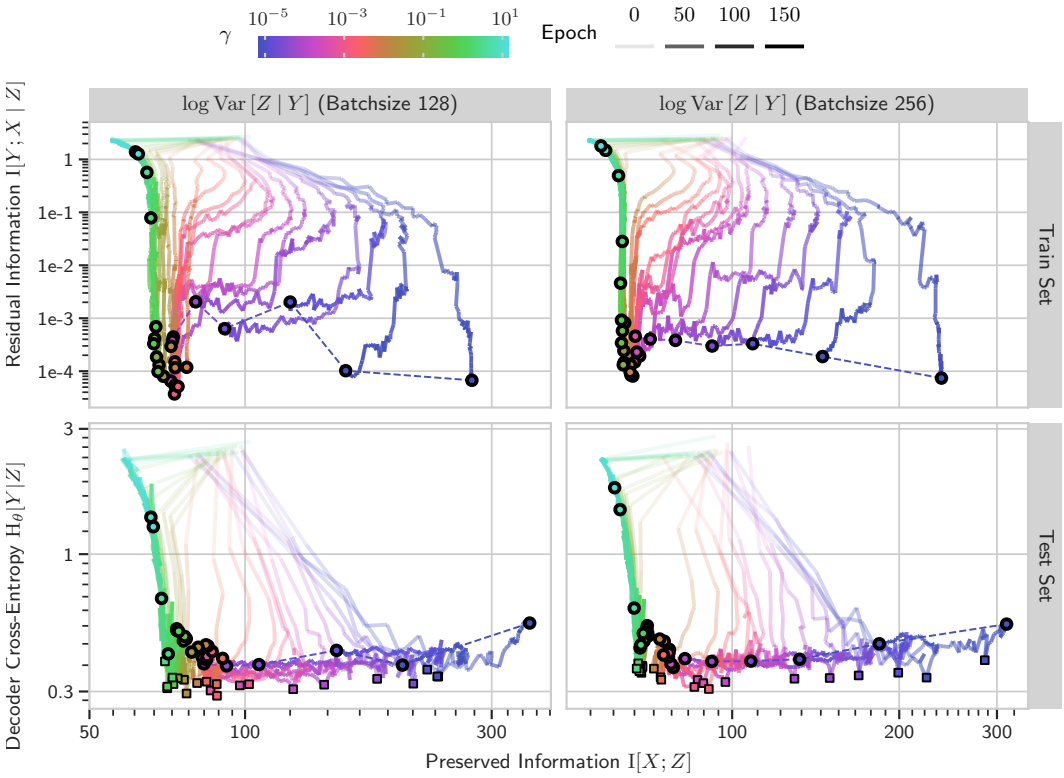


Figure G.8: Without dropout but with zero-entropy noise: *Information Plane Plot of training trajectories for ResNet18 models on CIFAR-10 and* log Var[Z | Y] *regularizer with batchsizes 128 and 256.* The trajectories are colored by their respective γ; their transparency changes by epoch. Compression (Preserved Information ↓) trades-off with performance (Residual Information ↓). See section 4. The circle marks the final epoch of a trajectory. The square marks the best epoch (Residual Information ⇊).

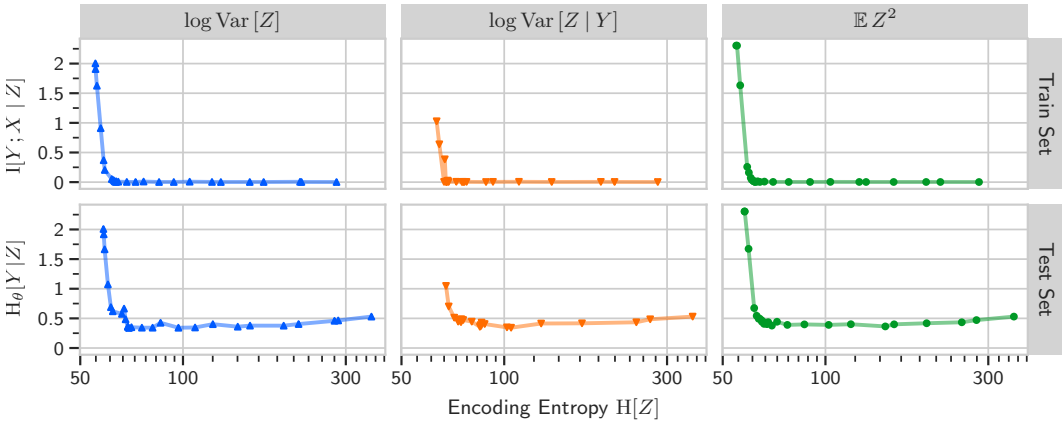
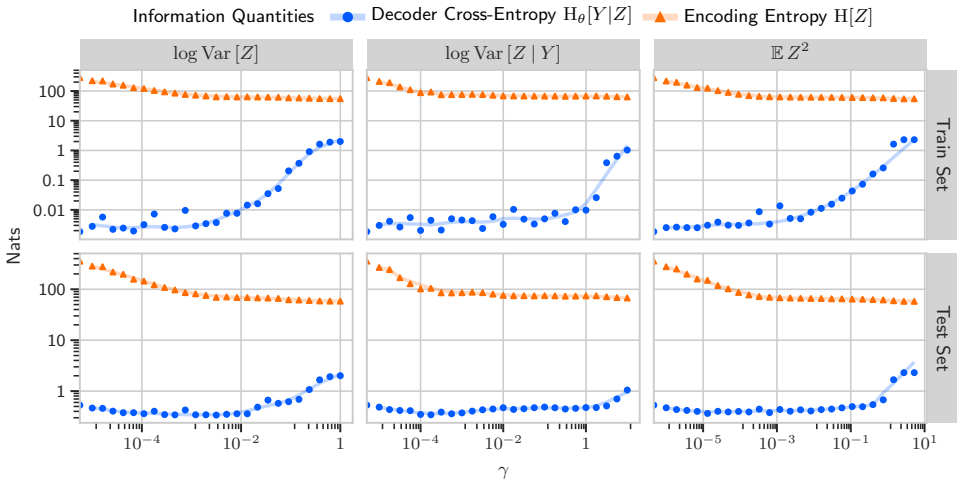
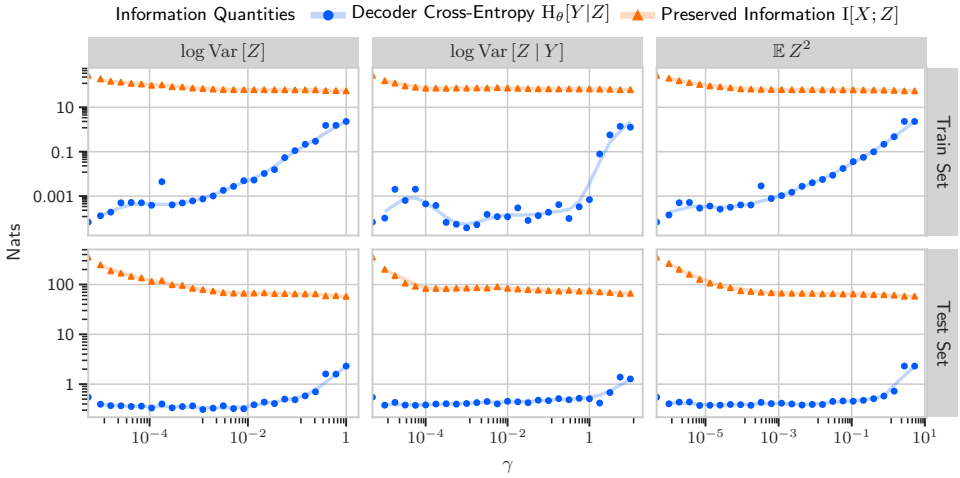


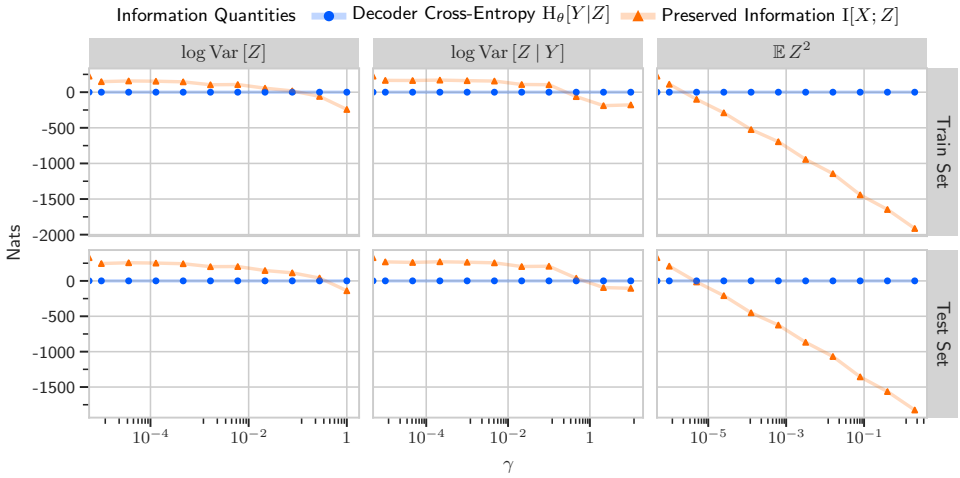
Figure G.9: *Information Plane Plot of the latent* Z *similar to Tishby and Zaslavsky (2015) but using a* ResNet18 *model on* CIFAR-10 *using the different regularizes from section 3.3 (*with Dropout and zero-entropy noise*).* The dots are colored by γ. See section 4 for more details.



(a) With dropout and zero-entropy noise.



(b) Without dropout but with zero-entropy noise.



(c) Without dropout and without zero-entropy noise.

Figure G.10: *Information quantites for different γ at the end of training for ResNet18 models on CIFAR-10 and* log Var[Z | Y] *regularizer with batchsizes 128 and 256.* Compression (Preserved Information ↓) trades-off with performance (Residual Information ↓). See section 4.

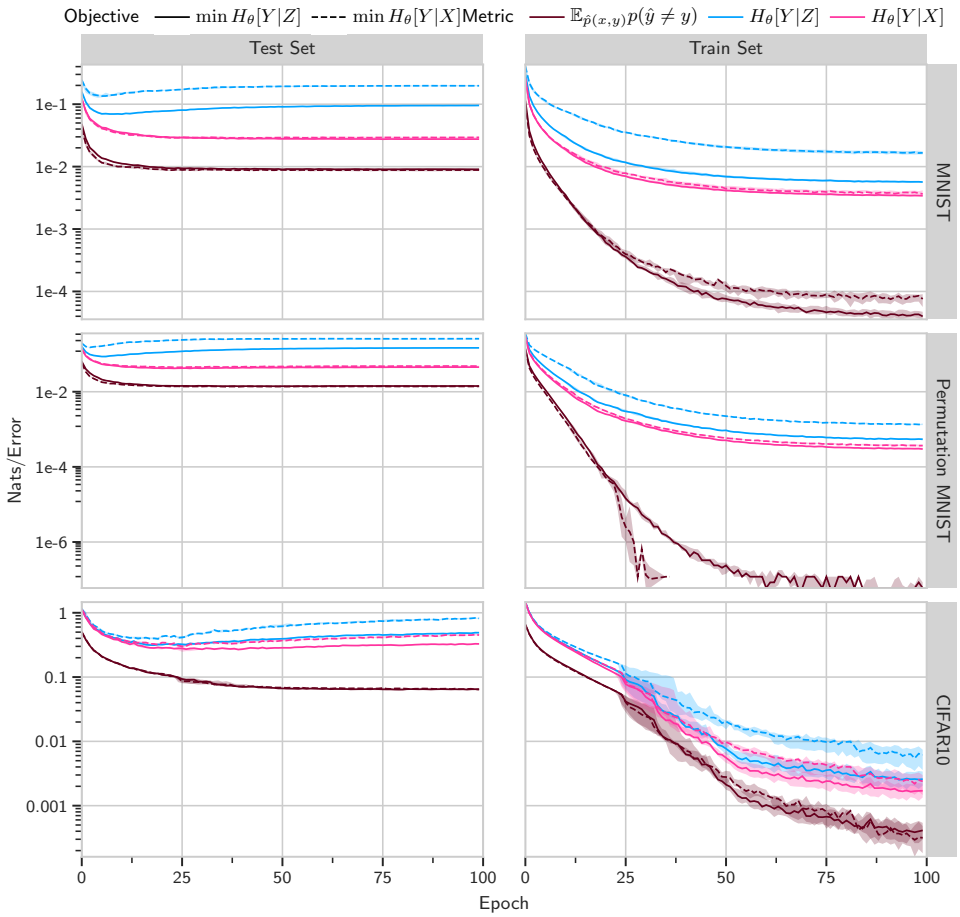


Figure G.11: *Training error probability, Decoder Cross-Entropy* $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$ *and Prediction Cross-Entropy* $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}]$ *with continuous* Z. $K = 100$ dimensions are used for Z, and we use Dropout to obtain stochastic models. Minimizing $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{Z}]$ (solid) leads to smaller cross-entropies and lower training error probability than minimizing $\mathrm{H}_\theta[\mathrm{Y} \mid \mathrm{X}]$ (dashed). This suggests a better data fit, which is what we desire for a loss term. We run 8 trials each and plot the median with confidence bounds (25% and 75% quartiles). See section 3.2 and G.3.3 for more details.

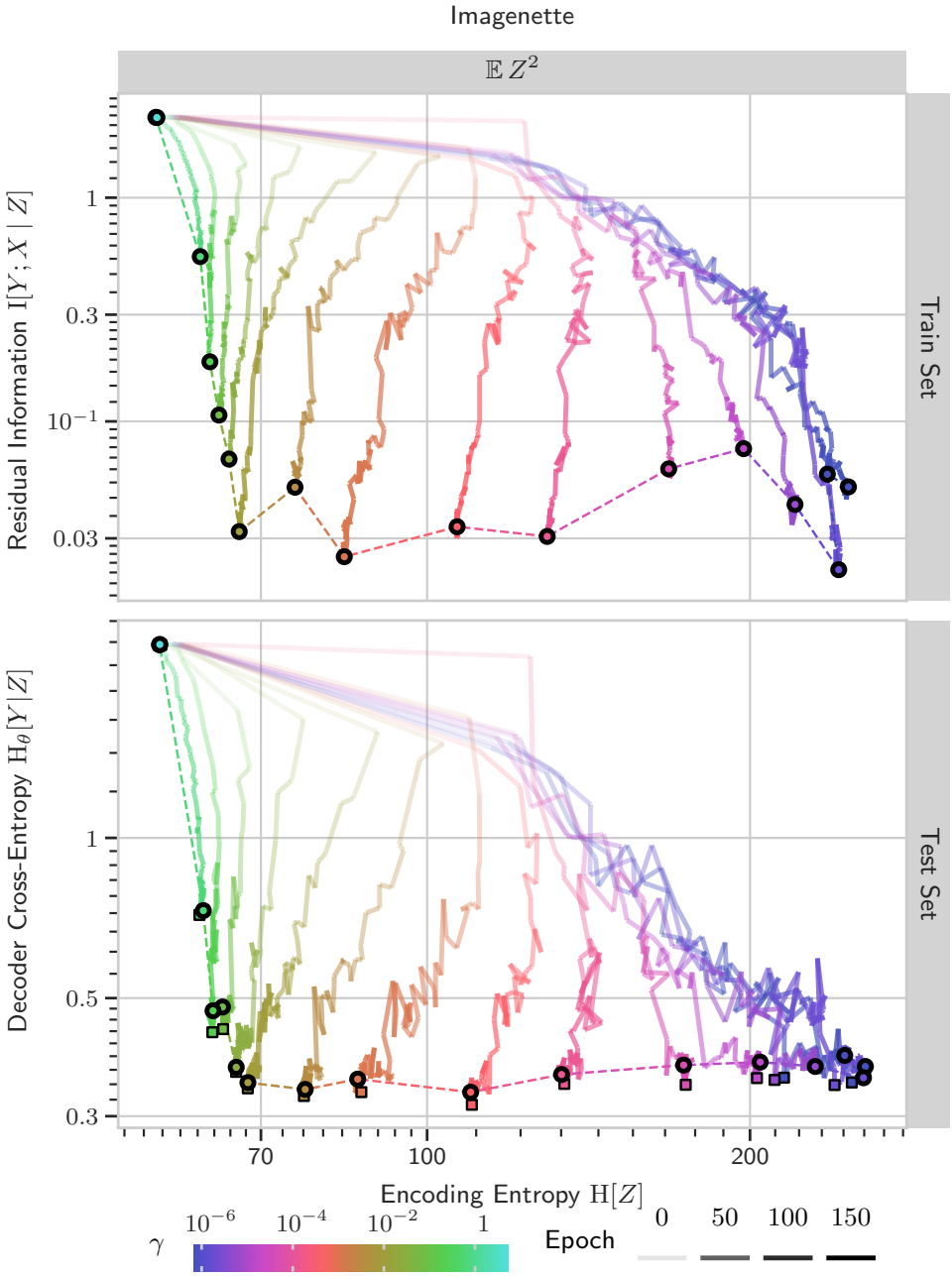


Figure G.12: *Information Plane Plot of the latent* Z *similar to Tishby and Zaslavsky (2015) but using a* ResNet18v2 *model on* ImageNette *using the* $\mathbb{E}\,\|Z\|^2$ *surrogate obejctive from section 3.3 (*with Dropout and zero-entropy noise*).* The dots are colored by γ.

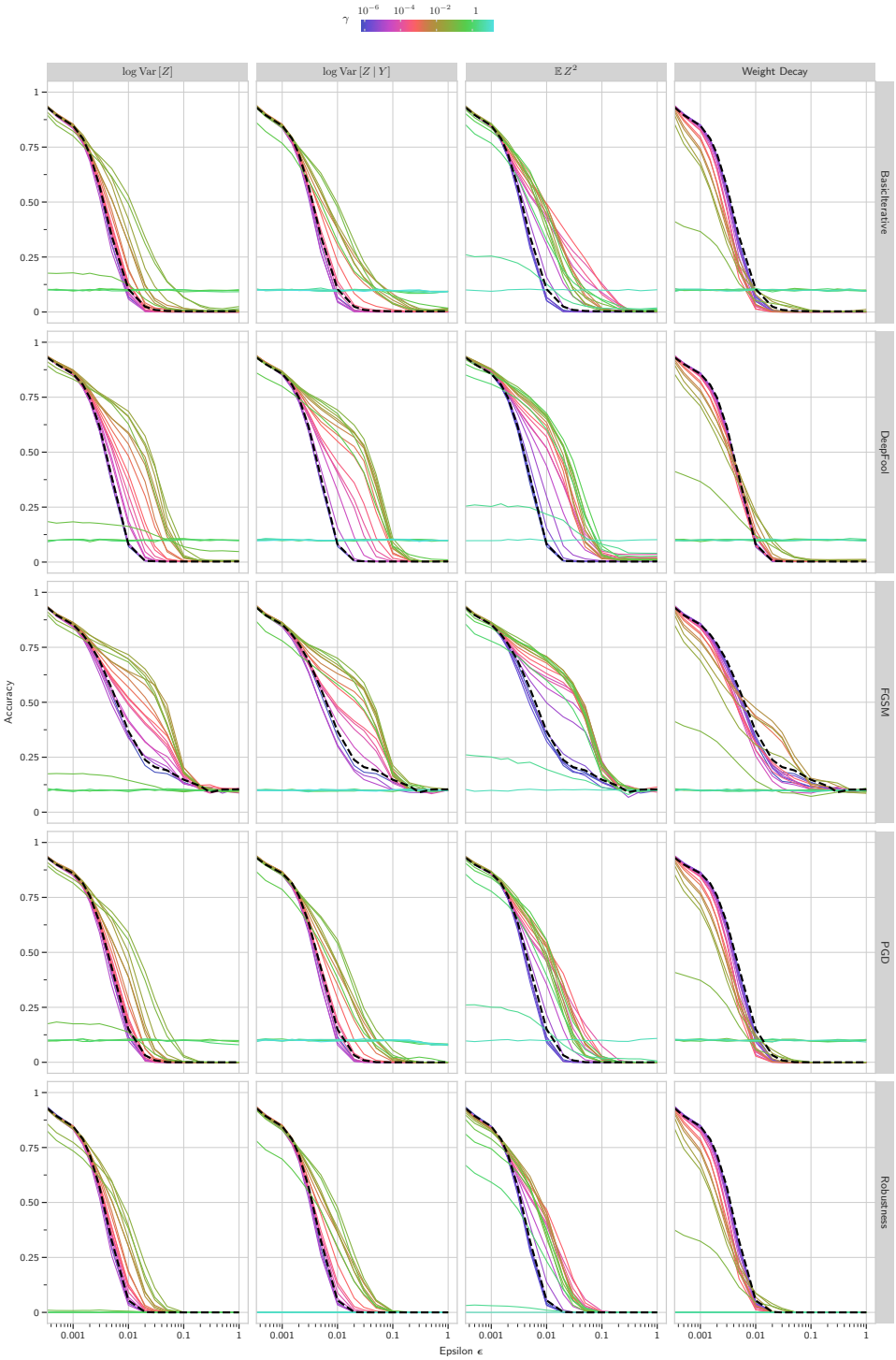


Figure G.13: *Adversarial robustness of ResNet18 models trained on CIFAR-10 with surrogate objectives in comparison to regularization with L2 weight-decay as non-IB method for different attack strengths ϵ.* The robustness is evaluated using FGSM, PGD, DeepFool and BasicIterative attacks of varying ϵ values. The dashed black line represents a model trained only with cross-entropy and no noise injection. We see that models trained with the surrogate IB objective (colored by γ) see improved robustness over a model trained only to minimize the cross-entropy training objective (shown in black) while the models regularized with weight-decay actually perform worse.

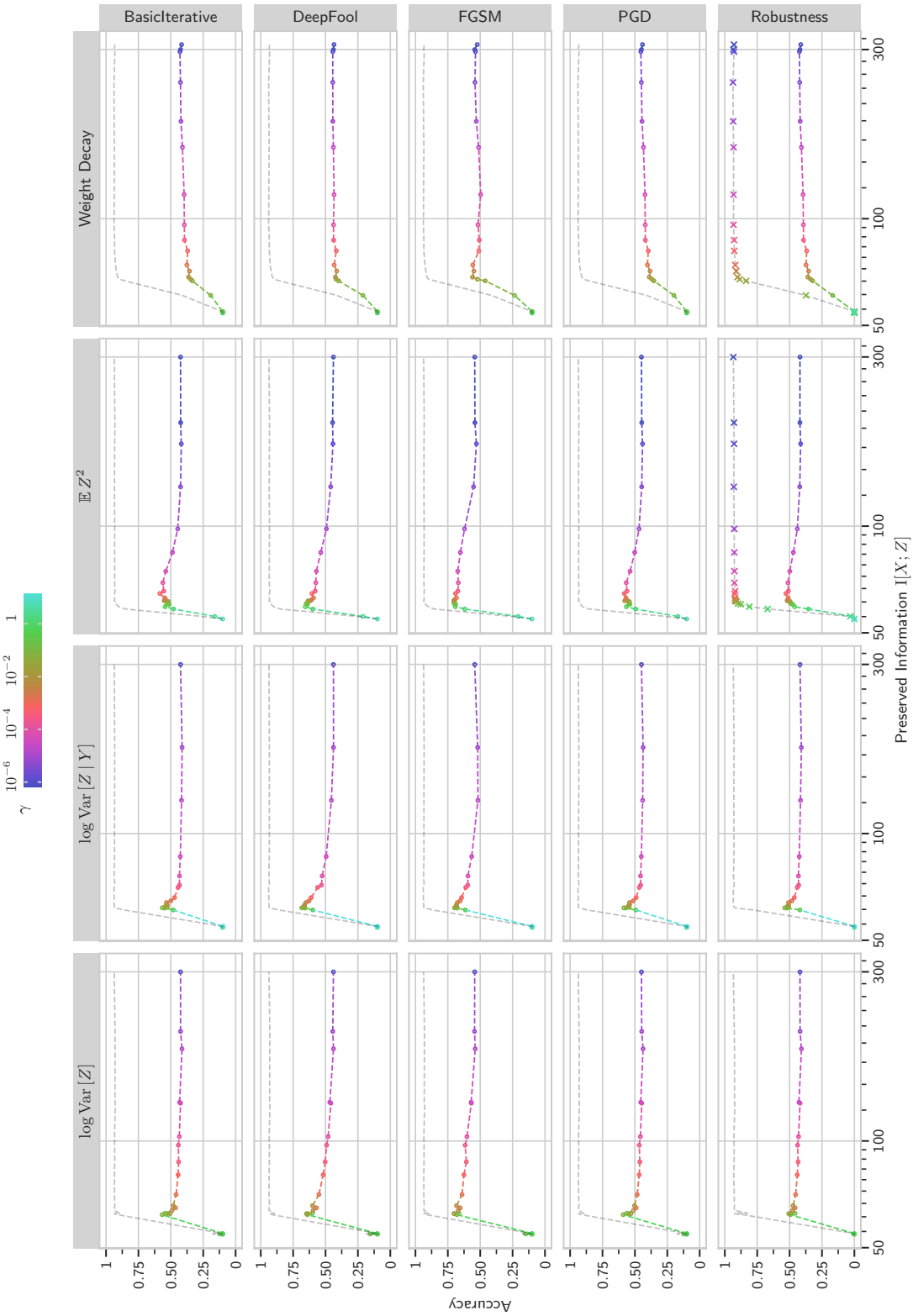


Figure G.14: *Average adversarial robustness over $\epsilon \in [0, 0.1]$ of ResNet18 models trained on CIFAR-10 with surrogate objectives in or L2 weight-decay (as non-IB method) compared to normal accuracy for different amounts of Preserved Information.* ○ markers show robustness. × markers show the normal accuracy. We see that robustness depends on the Preserved Information. If the latent is compressed too much, robustness (and accuracy) are low. If the latent is not compressed enough, robustness and thus generalization suffer.

H LARGE VERSION OF THE MICKEY MOUSE I-DIAGRAM

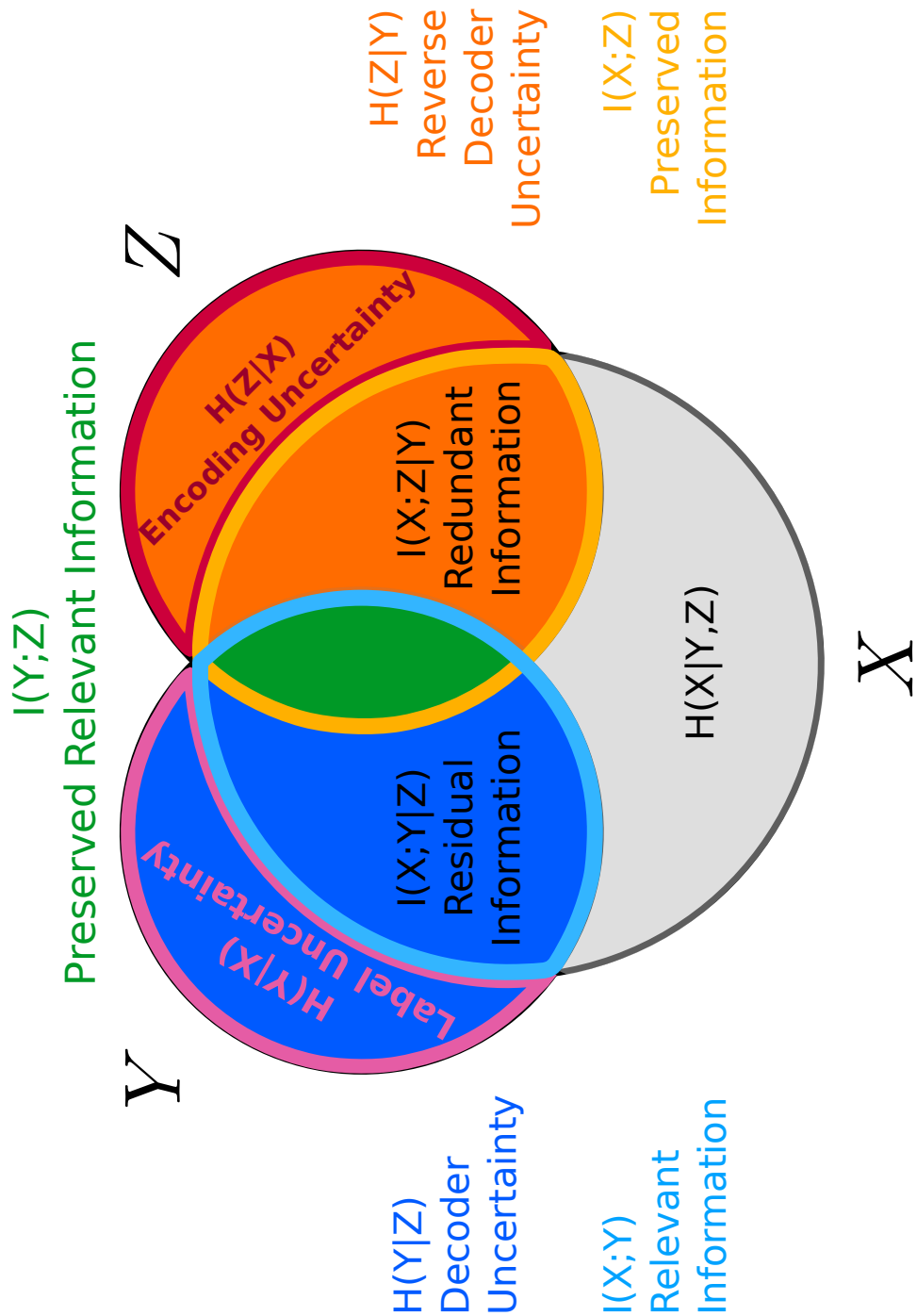


Figure H.1: *Mickey Mouse I-diagram.* See figure 2 for details.