

# Competing Mutual Information Constraints with Stochastic Competition-based Activations for Learning Diversified Representations

Konstantinos P. Panousis<sup>1\*</sup>, Anastasios Antoniadis<sup>1\*</sup>, Sotirios Chatzis<sup>1</sup>

<sup>1</sup>Department of Electrical Engineering and Computer Engineering and Informatics  
Cyprus University of Technology, Limassol, Cyprus  
k.panousis@cut.ac.cy

## Abstract

This work aims to address the long-established problem of learning diversified representations. To this end, we combine information-theoretic arguments with stochastic competition-based activations, namely Stochastic Local Winner-Takes-All (LWTA) units. In this context, we ditch the conventional deep architectures commonly used in Representation Learning, that rely on non-linear activations; instead, we replace them with sets of locally and stochastically competing linear units. In this setting, each network layer yields sparse outputs, determined by the outcome of the competition between units that are organized into blocks of competitors. We adopt stochastic arguments for the competition mechanism, which perform posterior sampling to determine the winner of each block. We further endow the considered networks with the ability to infer the sub-part of the network that is essential for modeling the data at hand; we impose appropriate stick-breaking priors to this end. To further enrich the information of the emerging representations, we resort to information-theoretic principles, namely the Information Competing Process (ICP). Then, all the components are tied together under the stochastic Variational Bayes framework for inference. We perform a thorough experimental investigation for our approach using benchmark datasets on image classification. As we experimentally show, the resulting networks yield significant discriminative representation learning abilities. In addition, the introduced paradigm allows for a principled investigation mechanism of the emerging intermediate network representations.

## 1 Introduction

The ability to extract diversified representations is the main focal point of Representation Learning (RL). Despite the immense amount of research effort, though, learning representations with such qualities remains an open research question. Recently, information-theoretic arguments, and specifically mutual information constraints (Bell and Sejnowski 1995; Belghazi et al. 2018; Hjelm et al. 2019a; van den Oord, Li, and Vinyals 2019), have risen as a promising direction towards successful RL. Notably, the Information Bottleneck (IB) (Tishby, Pereira, and Bialek 1999) has attracted a substantial amount of attention (Alemi et al. 2017; Achille and

Soatto 2017; Dai, Zhu, and Wipf 2018). The main IB principle lies on constraining the inferred representations, such that the information carried about the input is minimized, while being maximally informative about the target outputs (Tishby and Zaslavsky 2015). The first implementations of the IB principle in deep learning can be found in (Shwartz-Ziv and Tishby 2017; Alemi et al. 2017; Saxe et al. 2018).

Despite these advances, it is striking that the majority of the proposed works refrain from examining a core -and probably the most important- aspect of modern deep architectures; that is, the employed non-linearities. We posit that a radically different paradigm of latent unit operation may allow for significantly enhancing the representation power of deep networks. We draw inspiration from biologically-plausible architectures: it has been shown that neurons with similar functions in the mammal brain aggregate together in groups/blocks, and a local competition takes place for their activation (Kandel, Schwartz, and Jessell 2000; Andersen et al. 1969; Stefanis 1969; Douglas and Martin 2004; Lansner 2009). This leads to a Local Winner-Takes-All (LWTA) mechanism; the winner of the competition in each block gets to convey its activation outside the block, while the rest are inhibited to silence. The incorporation of the LWTA mechanism in deep architectures has been shown to exhibit significant properties such as *noise suppression*, *robustness to adversarial attacks* and *compression* (Srivastava et al. 2013; Grossberg 1982; Carpenter and Grossberg 1988; Panousis, Chatzis, and Theodoridis 2019, 2021; Panousis et al. 2021). Despite these properties, LWTA-based networks have been scarcely examined in the RL literature.

On this basis, in this work, we propose a novel deep learning design framework that: (i) replaces the commonly employed non-linear activations with *stochastic* LWTA activations; (ii) endows the considered networks with a data-driven *component utility inference mechanism*, which allows to infer the *essential* sub-parts of the network necessary for modeling the available data; and (iii) utilizes recent advances in the information-theoretic approaches to yield diversified representations, namely *competing mutual information constraints* (Hu et al. 2019). Our ultimate goal is to yield deep networks with considerably diversified resulting representations. We evaluate our approach using well-known benchmark datasets and architectures for image classification.

The rest of the paper is organized as follows: In Section 2,

\*Both authors contributed equally.

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

we introduce the foundation of our methodology. In Section 3, we formulate our novel representation method, and provide associated training and prediction algorithms. In Section 4, we perform rigorous experimental evaluations, investigate the behavior of the proposed framework, and provide strong separability results and insights into the emerging representations. Finally, in Section 5, we provide some final remarks.

## 2 Foundational Principles

Let us assume a dense layer of a conventional deep neural network comprising  $K$  hidden units. When presented with an input  $\mathbf{x} \in \mathbb{R}^J$ , each hidden unit  $k$  performs an inner product computation,  $h_k = \mathbf{w}_k^T \mathbf{x} = \sum_{j=1}^J w_{jk} \cdot x_j \in \mathbb{R}$ ;  $\mathbf{W} \in \mathbb{R}^{J \times K}$  is the associated weight matrix of the layer. This response usually passes through a non-linear function  $\sigma(\cdot)$ , yielding  $y_k = \sigma(h_k)$ ,  $\forall k$ . Thus, the final output of the layer arises from the concatenation of the non-linear activations of each unit, such that  $\mathbf{y} = [y_1, \dots, y_K] \in \mathbb{R}^K$ .

In contrast, in the LWTA framework, singular non-linear hidden units are replaced by  $U$  competing linear units grouped together in an (LWTA) block; each layer comprises multiple such blocks. Hereinafter, we denote by  $B$  the number of such blocks in a particular LWTA-based layer. The weight synapses are now structured as a three dimensional matrix  $\mathbf{W} \in \mathbb{R}^{J \times B \times U}$ , revealing that, in this case, each input is now presented to each block  $b$  and each unit  $u$  therein. In this context, each linear unit  $u$  in each block  $b$  computes its response, following the conventional inner product computation, such that  $h_{b,u} = \mathbf{w}_{b,u}^T \mathbf{x} = \sum_{j=1}^J w_{j,b,u} \cdot x_j \in \mathbb{R}$ ; then, competition takes places among the units in the block.

The main operating principle is that out of the  $U$  units in the block, *only one can be the winner*; this unit gets to convey its (linear) activation to the next layer, while *all the rest are inhibited to silence*, i.e. pass a zero value. The final output of an LWTA-based layer  $\mathbf{y} \in \mathbb{R}^{B \times U}$  is now composed of  $B$  subvectors  $\mathbf{y}_b \in \mathbb{R}^U$ , one for each LWTA block and each with a *single non-zero entry*. It is evident that this process results in a *sparse representation*; in each block, only one out of the  $U$  units produces a non-zero output<sup>1</sup>. In the related LWTA literature, the competition process is usually deterministic, i.e., the winner with the *highest linear activation* is deemed the winner each time. However, novel data-driven stochastic arguments for the competition process have been recently proposed in (Panousis, Chatzis, and Theodoridis 2019, 2021; Panousis et al. 2021; Voskou et al. 2021).

In this setting, to encode the outcome of the competition in each of the  $B$  stochastic LWTA blocks that constitute a stochastic LWTA layer, we introduce an appropriate set of discrete latent vectors  $\xi \in \text{one\_hot}(U)^B$ . This vector comprises  $B$  component subvectors; each component entails *exactly one non-zero value* at the *index position* that corresponds to the *winner unit* in each respective LWTA block.

Further, in this work, we introduce an additional data-driven mechanism in order to endow the networks with the

<sup>1</sup>The higher the number of competitors  $U$  in each block, the sparser the output. When  $U = 2$ , only 50% of the units in a layer are active for each example, when  $U = 4$ , only 25%, e.t.c.

ability to *infer* which subparts of the network are *essential* for modeling the data at hand. To this end, we introduce a matrix of auxiliary binary latent variables  $\mathbf{Z} \in \{0, 1\}^{J \times B}$ . Each entry therein is 1 if the  $j^{\text{th}}$  component of the input is *presented* to the  $b^{\text{th}}$  block and zero otherwise. In this setting, the response of each unit is facilitated via the inner product operation between the *effective* network weights (as dictated by the latent indicators  $\mathbf{Z}$ ) and the input.

We can now express the output  $\mathbf{y}$  of a stochastic LWTA layer's  $(b, u)^{\text{th}}$  unit,  $y_{b,u}$ , that is, the output of the  $u^{\text{th}}$  unit in the  $b^{\text{th}}$  block, as:

$$y_{b,u} = \xi_{b,u} \sum_{j=1}^J (z_{j,b} \cdot w_{j,b,u}) \cdot x_j \in \mathbb{R} \quad (1)$$

where  $\xi_{b,u}$  denotes the  $u^{\text{th}}$  component of  $\xi_b$ , and  $\xi_b \in \text{one\_hot}(U)$  holds the  $b^{\text{th}}$  subvector of  $\xi$ .

We postulate that the winner unit latent indicators  $\xi_b$ ,  $\forall b$  in Eq. (1) are obtained via a *competitive* random sampling procedure; this, translates to drawing samples from a Categorical distribution, where the probabilities are proportional to the intermediate linear computation that each unit performs. Accordingly, the higher the linear response of a particular unit in a particular block, the higher its probability of being the winner in said block; this yields:

$$q(\xi_b) = \text{Categorical}\left(\xi_b \mid \text{softmax}\left(\sum_{j=1}^J z_{j,b} \cdot [w_{j,b,u}]_{u=1}^U \cdot x_j\right)\right) \quad (2)$$

where  $[w_{j,b,u}]_{u=1}^U$  denotes the vector concatenation of the set  $\{w_{j,b,u}\}_{u=1}^U$ . On the other hand, we postulate that the binary latent indicators  $\mathbf{Z}$  are drawn from a Bernoulli distribution, operating in an “on”–“off” fashion, such that:

$$q(z_{j,b}) = \text{Bernoulli}(z_{j,b} | \tilde{\pi}_{j,b}) \quad (3)$$

where  $\tilde{\pi}_{j,b}$ ,  $\forall j, b$  are trainable parameters. A graphical illustration of the proposed stochastic LWTA block is depicted in Fig.1. Each stochastic LWTA layer comprises multiple such LWTA blocks as illustrated in Fig. 2a. At this point, it is important to note a key aspect of the proposed approach; that is, *stochasticity*. In each layer, different stochastic representations arise due to the sampling procedure for both latent indicators  $\xi$  and  $\mathbf{Z}$ . Even when presented with the same input, different *subnetworks* may be activated and different *subpaths* are followed from the input to the output, as a result of winner,  $\xi$ , and component utility,  $Z$ , sampling.

## A Convolutional Variant

To account for networks employing the convolutional operation, popular in RL, we formulate a convolutional variant of the proposed Stochastic LWTA rationale. Let us now consider an input tensor  $\mathbf{X} \in \mathbb{R}^{H \times L \times C}$ , where  $H, L, C$  are the height, length and channels of the input. In this context, we define a set of kernels, each with weights  $\mathbf{W}_b \in \mathbb{R}^{h \times l \times C \times U}$ , where  $h, l, C, U$  are the kernel height, length, channels and number of *competing feature maps*; each layer comprises  $B$  kernels. Analogously to the grouping and competition of linear units in dense layers, in this case, local competition

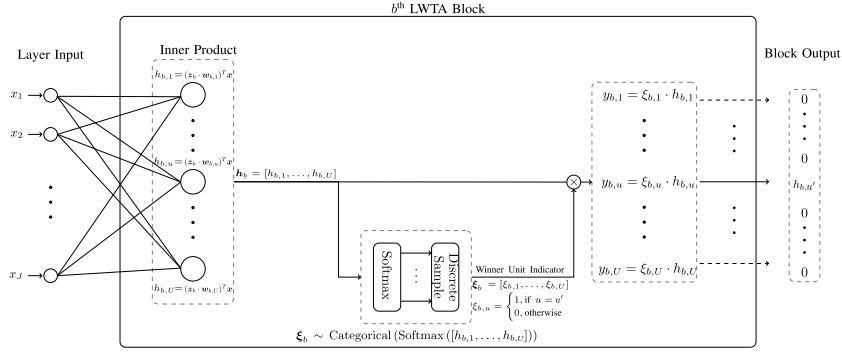


Figure 1: A detailed bisection of the  $b^{\text{th}}$  Stochastic LWTA block in an LWTA layer. Presented with an input  $\mathbf{x} \in \mathbb{R}^J$ , each unit  $u = 1, \dots, U$  computes its activation  $h_{b,u}$  via different weights  $\mathbf{w}_{b,u} \in \mathbb{R}^J$ , i.e.,  $h_{b,u} = (\mathbf{z}_b \cdot \mathbf{w}_{b,u}^T) \mathbf{x}$ . Here,  $\mathbf{z}_b$  is the component utility indicator pertaining to the  $b^{\text{th}}$  block, which encodes which synapses leading to the  $b^{\text{th}}$  block the inference algorithm deems useful, and which not. The linear responses of the units are concatenated, such that  $\mathbf{h}_b = [h_{b,1}, \dots, h_{b,U}]$ , and transformed into probabilities via the softmax operation. Then, a Discrete sample  $\xi_b = [\xi_{b,1}, \dots, \xi_{b,U}]$  is drawn; this constitutes a one-hot vector with a single non-zero entry at position  $u'$ , denoting the winner unit in the block. The winner unit passes its linear response to the next layer; the rest pass zero values.

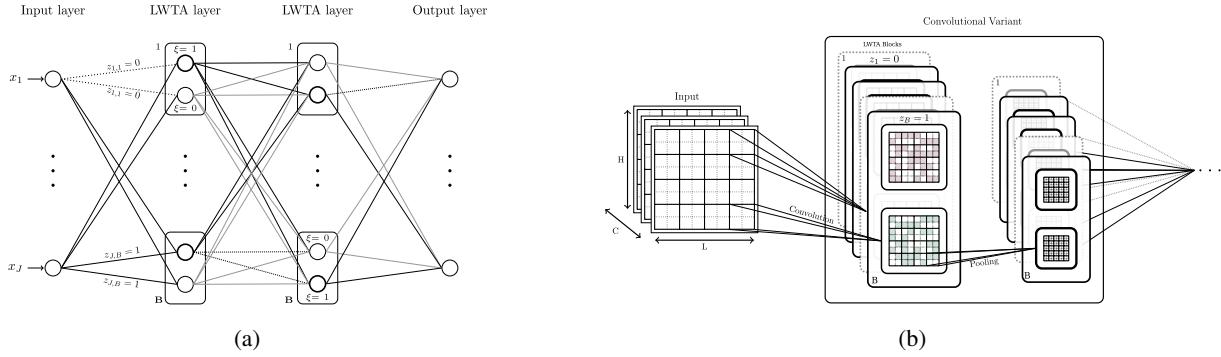


Figure 2: (a) A graphical representation of our competition-based modeling approach. Rectangles denote LWTA blocks, and circles the competing units therein. The winner units are denoted with bold contours ( $\xi = 1$ ). Bold edges denote retained connections ( $z = 1$ ). (b) The convolutional LWTA variant. Competition takes place among feature maps on a position-wise basis. The winner feature map at each position passes its output to the next layer, while the rest pass zero values at said position.

is performed on a *position-wise* basis among feature maps. Each kernel is treated as an LWTA block with *competing feature maps*. The feature maps of each kernel compete to win the activation pertaining to each position; thus, we have as many competition outcomes within the kernel as the number of positions (in the definition of the feature maps). This way, the latent winner indicator variables now pertain to the selection of winner feature maps on a *position-wise* basis.

Further, we introduce analogous latent utility indicators  $\mathbf{z} \in \{0, 1\}^B$  in order to infer which *kernels* (LWTA blocks) are necessary for modeling the available data. Thus, here, if  $z_b = 0$ , we omit *whole blocks of competing feature maps*. Under this regard, each feature map  $u = 1, \dots, U$  in the  $b^{\text{th}}$  LWTA block (kernel) of a convolutional LWTA layer computes:

$$\mathbf{H}_{b,u} = (\mathbf{z}_b \cdot \mathbf{W}_{b,u}) * \mathbf{X} \in \mathbb{R}^{H \times L} \quad (4)$$

Then, competition takes place among the  $U$  kernel feature maps for claiming the available *positions*, one by one. Specif-

ically, the competitive random sampling procedure reads:

$$q(\xi_{b,h',l'}) = \text{Categorical}\left(\xi_{b,h',l'} \mid \text{softmax}\left([\mathbf{H}_{b,u,h',l'}]_{u=1}^U\right)\right) \quad (5)$$

where  $[\mathbf{H}_{b,u,h',l'}]_{u=1}^U$  denotes the concatenation of the set  $\{\mathbf{H}_{b,u,h',l'}\}_{u=1}^U$ . In each kernel  $b = 1, \dots, B$ , and for each position  $h' = 1, \dots, H$ ,  $l' = 1, \dots, L$ , only the winner feature map contains a non-zero entry; all the rest feature maps contain zero values at these positions. This yields sparse feature maps with mutually exclusive active pixels. Accordingly for the utility indicators:

$$q(z_b) = \text{Bernoulli}(z_b | \tilde{\pi}_b), \forall b \quad (6)$$

Thus, the output  $\mathbf{Y} \in \mathbb{R}^{H \times L \times B \cdot U}$  of a convolutional layer of the proposed stochastic LWTA-based networks is obtained via concatenation of the subtensors  $\mathbf{Y}_{b,u}$  that read:

$$\mathbf{Y}_{b,u} = \Xi_{b,u}((z_b \cdot \mathbf{W}_{b,u}) * \mathbf{X}), \forall b, u \quad (7)$$

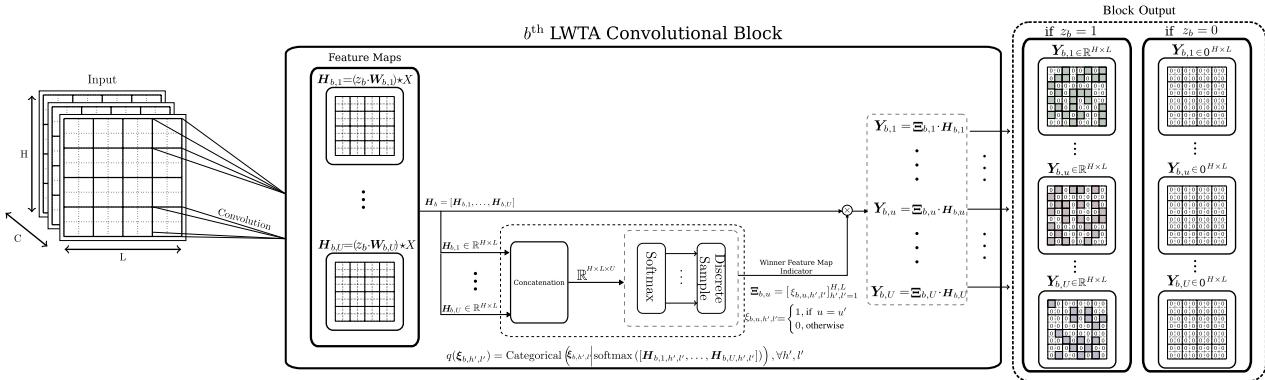


Figure 3: A detailed bisection of the  $b^{\text{th}}$  convolutional stochastic LWTA block. Presented with an input  $\mathbf{X} \in \mathbb{R}^{H \times L \times C}$ , competition now takes place among feature maps on a position-wise basis. Only the winner feature map contains a non-zero entry in a specific position. This leads to sparse feature maps, each comprising uniquely position-wise activated pixels. The component utility indicators  $z_b$  encode which kernels/blocks the algorithm deems useful; when  $z_b = 0$ , whole blocks are omitted.

where  $\Xi_{b,u} = [\xi_{b,u,h',l'}]_{h',l'=1}^{H,L}$ . The corresponding illustration of the proposed stochastic convolutional LWTA block is depicted in Fig. 3. Convolutional stochastic LWTA-based layers comprise multiple such blocks, as shown in Fig. 2b.

### 3 Representation Leaning & Mutual Information

Mutual information (MI) has been an indispensable tool in RL (Bell and Sejnowski 1995; Linsker 1988; van den Oord, Li, and Vinyals 2019). In this context, (Tishby, Pereira, and Bialek 1999) introduced the Information Bottleneck (IB); this employs an information-theoretic objective that constrains the MI between the input and an intermediate representation, while enhancing the MI between said representation and the output (Tishby and Zaslavsky 2015; Shwartz-Ziv and Tishby 2017; Saxe et al. 2018; Dai, Zhu, and Wipf 2018).

In this line of work, the recently proposed Deep Variational Information Bottleneck (VIB) (Alemi et al. 2017) constitutes a variational approximation to the IB objective, yielding networks with increased generalization and robustness capabilities. Notably, there exists a connection between VIB and the well-known  $\beta$ -VAE formulation (Alemi, Fischer, and Dillon 2018; Alemi et al. 2018). Both are founded on information theoretic arguments but used in different contexts; the first for supervised while the latter for unsupervised learning (Higgins et al. 2017). Here, we focus on the former.

Despite significant progress in the field, the majority of existing methods optimize a single information constraint to learn “useful” and diverse representations; however, this mode of operation fails to promote diversity among the representations inferred from the latent units.

Thus, various learning schemes have been devised in the literature to address this inadequacy; these are based on the idea of collaboration and competition between different neural representations (Greff et al. 2016; Greff, van Steenkiste, and Schmidhuber 2017; Ranjan et al. 2019). On this basis, (Hu et al. 2019) recently proposed the *information-competing process* (ICP) scheme; this construct entails different repre-

sentation parts that not only *compete*, but also *collaborate* to achieve a downstream task, thus enabling the model to learn richer and more discriminative representations.

### Proposed Approach

Let us denote by  $t$ , the output of the *downstream task*, and by  $\mathbf{r}$  the learned representation of  $\mathbf{x}$ ; in supervised learning,  $t$  corresponds to the target label pertaining to an observation  $\mathbf{x}$ . Most information-theoretic approaches aim to maximize a single constraint, e.g., the MI between the representation  $\mathbf{r}$  and target  $t$  denoted as  $\mathcal{I}(\mathbf{r}, t)$ . In contrast, ICP takes a different route. Initially, in order to promote diversification of the emerging representations,  $\mathbf{r}$  is *explicitly split* into two different parts,  $\{\zeta, \mathbf{y}\}$ ; each is imposed different constraints: the “information capacity” of  $\zeta$  is minimized, while the “information capacity” of  $\mathbf{y}$  is maximized. Under this regard, the objective reads:

$$\max [\mathcal{I}(\mathbf{r}, t) + \alpha \mathcal{I}(\mathbf{y}, \mathbf{x}) - \beta \mathcal{I}(\zeta, \mathbf{x})] \quad (8)$$

where  $\alpha, \beta$  are regularization constants.  $\mathcal{I}(\zeta, \mathbf{x})$  and  $\mathcal{I}(\mathbf{y}, \mathbf{x})$  are called the *mutual information minimization* and *maximization* terms respectively. At this point, it is important to note that computing the MI between different pairs of variables is usually intractable; thus, we need to devise different optimization schemes for each considered term. For example, to minimize  $\mathcal{I}(\zeta, \mathbf{x})$ , we introduce a *variational approximation*  $Q(\zeta)$  to  $P(\zeta)$ , yielding the following tractable upper bound:

$$\mathcal{I}(\zeta, \mathbf{x}) \leq \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\text{KL}[P(\zeta|\mathbf{x})||Q(\zeta)]] \quad (9)$$

where  $Q(\zeta)$  is a standard Gaussian distribution. The parameters of this distribution, i.e.,  $\mu_\zeta, \sigma_\zeta$ , are founded on an amortization scheme, similar to the well-known VAE formulation (Kingma and Welling 2014); hence, deep neural networks are employed for their estimation.

In this context, before proceeding to define the full ICP objective, we *highlight* that, *all components* stemming from the corresponding tractable bounds and optimization schemes described next, e.g., feature extractors, discriminators and

classifiers are originally implemented in (Hu et al. 2019) via ReLU-based DNNs. On this basis, this work introduces a radically different view: We employ deep networks built of novel *stochastic LWTA* arguments, instead of standard nonlinearities. Further, we complement this unique latent unit operation with a sparsity-inducing framework that allows determining the best postulated sub-network configuration.

As already discussed, a notable aspect of ICP is the introduction of auxiliary constraints in order to prevent either part dominating the downstream task. To this end, the separated parts are individually allowed to accomplish  $t$  via dedicated MI constraints, namely  $\mathcal{I}(\zeta, t)$  and  $\mathcal{I}(y, t)$ ; however,  $\zeta$  and  $y$  are *constrained from knowing what each other has learned*. This property is realized via an additional constraint, that is by minimizing  $\mathcal{I}(\zeta, y)$ ; this forces  $\zeta$  and  $y$  to be independent of each other. The so-obtained objective reads:

$$\begin{aligned}\mathcal{L}_{\text{ICP}} = \max & \left[ \mathcal{I}(r, t) + \alpha \mathcal{I}(y, x) - \beta \mathcal{I}(x, \zeta) \right. \\ & \left. + \mathcal{I}(\zeta, t) + \mathcal{I}(y, t) - \gamma \mathcal{I}(\zeta, y) \right]\end{aligned}\quad (10)$$

where  $\gamma$  is another regularization constant,  $\mathcal{I}(\zeta, t)$ ,  $\mathcal{I}(y, t)$ ,  $\mathcal{I}(r, t)$  are called the *inference* terms, and  $\mathcal{I}(\zeta, y)$  the *predictability minimization* term.  $\mathcal{I}(r, t)$  denotes the “synergy” between the separate parts, aiming to accomplish the downstream task in a *synergistic way*. Contrarily, the last three terms, i.e.,  $\mathcal{I}(\zeta, t)$ ,  $\mathcal{I}(\zeta, y)$ ,  $\mathcal{I}(y, t)$  constitute the “competition” aspect of the approach. This conception allows for both competition as well as synergy of the different representation parts, enhancing the information carried by said representations. In the following, we briefly present the required optimization schemes for each term of the objective. In the case of  $\mathcal{I}(y, x)$  the KL divergence is divergent (Hu et al. 2019); thus, we resort to maximization of the Jensen-Shannon (JS) divergence. Its variational estimation yields:

$$\begin{aligned}\text{JS}[P(y|x)P(x)||P(y)P(x)] = & \\ \max & \left[ \mathbb{E}_{(y,x) \sim P(y|x)p(x)} [\log D(y, x)] \right. \\ & \left. + \mathbb{E}_{(\hat{y},x) \sim P(y)p(x)} [\log(1 - D(\hat{y}, x))] \right]\end{aligned}\quad (11)$$

where  $D(\cdot)$  is a discriminator, estimating the probability of an input pair;  $(y, x)$  is the positive pair sampled from  $P(y|x)P(x)$  and  $(\hat{y}, x)$  is the negative pair sampled from  $P(y)P(x)$ ;  $\hat{y}$  is a “disorganized” version of  $y$  (Hu et al. 2019). For the *inference* term  $\mathcal{I}(r, t)$ , the following lower bound is derived:

$$\mathcal{I}(r, t) \geq \mathbb{E}_{x \sim P(x)} \left[ \mathbb{E}_{r \sim P(r|x)} \left[ \int P(t|x) \log Q(t|r) dt \right] \right] \quad (12)$$

where  $Q(t|r)$  is a variational approximation of  $P(t|r)$ , and  $P(t|x)$  denotes the distribution of the labels. Thus, Eq.(12) essentially constitutes the cross-entropy loss. The expressions for the inference terms  $\mathcal{I}(\zeta, t)$  and  $\mathcal{I}(y, t)$  are similar.

Finally, we turn to the *predictability minimization term*; this enforces the representations parts to be independent. To this end, we introduce a predictor  $H$  that is used to predict  $y$  given  $\zeta$ ; this way, we guide the  $\zeta$  variable extractor from producing  $\zeta$  values that can predict  $y$  (Hu et al. 2019). The same procedure is followed for  $y$  to  $\zeta$ . This yields:

$$\min \max \left[ \mathbb{E}_{\zeta \sim P(\zeta|x)} [H(y|\zeta)] + \mathbb{E}_{y \sim P(y|x)} [H(\zeta|y)] \right] \quad (13)$$

A graphical illustration of the described optimization process of the ICP objective of Eq.(10) is depicted in Fig. 4. Therein, all DNN-based components are founded on the Stochastic LWTA and component utility arguments of Section 2.

## Training & Prediction

The core training objective was defined in the previous section; this stems from the ICP rationale as expressed in Eq.(10); however, the existence of Stochastic LWTA activations and the component utility mechanism, necessitates the augmentation of the final objective via appropriate KL terms.

Without loss of generality, we begin by considering a symmetric Categorical distribution for the latent variable indicators  $\xi$ ; hence,  $p(\xi_b) = \text{Categorical}(1/U) \forall b$  for the dense layers, and  $p(\xi_{b,h,l'}) = \text{Categorical}(1/U)$ ,  $\forall b, h, l'$  for convolutional ones. Differently, for the latent utility indicators  $Z$  (or  $z$ ) we do not impose a symmetric prior; instead, we turn to the non-parametric Bayesian framework and specifically to the Indian Buffet Process (IBP) (Ghahramani and Griffiths 2006). This constitutes a sparsity-promoting prior; at the same time, its so-called *stick-breaking process* (SBP) (Teh, Grür, and Ghahramani 2007) renders IBP amenable to Variational Inference. The hierarchical construction reads:

$$p(z_{j,b}) = \text{Bernoulli}(\pi_b), \quad \pi_b = \prod_{i=1}^b u_b, \quad u_b \sim \text{Beta}(\omega, 1) \quad (14)$$

where  $\omega$  is a non-negative constant, controlling the induced sparsity. The SBP requires an additional set of latent *stick* variables  $u_b, \forall b$ . Since these are Beta-distributed, we assume a posterior of similar form:  $q(u_b) = \text{Beta}(u_b | \tilde{a}_b, \tilde{b}_b)$ , where  $\tilde{a}_b, \tilde{b}_b, \forall b$  are trainable variational parameters. This yields:

$$\begin{aligned}\mathcal{L} = \mathcal{L}_{\text{ICP}} - \text{KL}[q(\xi) || p(\xi)] - \text{KL}[q(Z) || p(Z)] \\ - \text{KL}[q(u) || p(u)]\end{aligned}\quad (15)$$

For training, we perform Monte-Carlo sampling to estimate Eq. (15) using a single reparameterized sample for each latent variable. For the Gaussian distributed variables, e.g.  $\zeta$ , we resort to the well-known Gaussian trick. For  $Z$  and  $u$ , these are obtained via the continuous relaxation of the Categorical and Bernoulli distribution (Jang, Gu, and Poole 2017; Maddison, Mnih, and Teh 2017). In the following, we describe the reparameterization trick for the  $\xi$  variables of dense layers; the cases for  $Z$  and for the convolutional variant are analogous.

Let  $\tilde{\xi}$  denote the probabilities of  $q(\xi)$  (Eqs.(2),(5)). Then, the samples  $\hat{\xi}$  can be expressed as:

$$\hat{\xi}_{b,u} = \text{Softmax}((\log \tilde{\xi}_{b,u} + g_{b,u})/\tau), \quad \forall b, u \quad (16)$$

where  $g_{b,u} = -\log(-\log V_{b,u})$ ,  $V_{b,u} \sim \text{Uniform}(0, 1)$  and  $\tau \in (0, \infty)$  is a *temperature* constant that controls the degree of the approximation. Similarly, the Beta distribution of the stick variables  $u$  is not readily amenable to reparameterization; for these, we obtain the required samples via the Kumaraswamy distribution (Kumaraswamy 1980); this constitutes an approximation of Beta and admits the following reparameterization trick:

$$\hat{u}_b = \left( 1 - (1 - G)^{\frac{1}{\tilde{a}_b}} \right)^{\frac{1}{\tilde{b}_b}} \quad (17)$$

where  $G \sim \text{Uniform}(0, 1)$  and  $\tilde{a}_b, \tilde{b}_b$  are the variational parameters of the original Beta distribution. We can now compute each expectation term in the objective (Eq.(15)) via these samples. For example, we can write the KL divergence term for the latent variables  $\xi$  as:

$$\begin{aligned} \text{KL}[q(\xi_b) || p(\xi_b)] &= \mathbb{E}_{q(\xi_b)} [\log q(\xi_b) - \log p(\xi_b)] \\ &\approx \log q(\hat{\xi}_b) - \log p(\hat{\xi}_b), \forall b \end{aligned} \quad (18)$$

At prediction time, we directly draw samples from the trained posteriors  $q(\xi)$  and  $q(Z)$  in order to determine the winner in each block of the network and to assess component utility respectively. Thus, each time we sample, even for the same input, a different *subpath* may be followed according to the outcomes of the sampling processes. This leads to a stochastic alternation of the emerging representations of the network at each forward pass.

## 4 Experimental Evaluation

We evaluate our model on image classification, where we consider two popular benchmarks, namely CIFAR-10 and CIFAR-100 (Krizhevsky 2009), containing natural images with 10 and 100 classes respectively. We compare our approach to recent information-theoretic approaches to deep networks, including VIB (Alemi et al. 2017), DIM (Hjelm et al. 2019a) and ICP (Hu et al. 2019). To this end, we follow the same experimental setup as (Hu et al. 2019), and consider four different networks for both datasets: (i) VGG-16 (Simonyan and Zisserman 2015), (ii) GoogLeNet (Szegedy et al. 2015), (iii) ResNet20 (He et al. 2016), and (iv) DenseNet40 (Huang et al. 2017). For both datasets, we normalize the raw image values using the per-channel means and standard deviations. We employ some common data augmentation procedures, including random cropping and mirroring.

## Experimental Setup

To evaluate our approach, we consider two different setups for our competition-based LWTA activations: (i) an architecture comprising LWTA blocks with  $U = 2$  competing units/feature maps, denoted as  $\text{ICP}_{\text{LWTA-2}}$ ; and (ii) an architecture comprising LWTA blocks with  $U = 4$  competing units/feature maps, denoted as  $\text{ICP}_{\text{LWTA-4}}$ . In all cases, the total number of hidden units/features in each layer remain the same as in the original ReLU-based architectures. This ensures comparability (size-wise) among existing approaches.

We also compare performance when we employ the proposed, IBP-driven, network sampling mechanism, and when we omit it. For clarity, in the following, we denote our full model (which employs the IBP-based mechanism) as  $\text{ICP}_{\text{IBP} \& \text{LWTA-2}}$  and  $\text{ICP}_{\text{IBP} \& \text{LWTA-4}}$ , respectively. In this context, to allow for compressing the model by exploiting the trained posteriors over the latent indicators  $z$ , we adopt the following rationale: After network training, we introduce a cut-off threshold  $\tau = 0.001$ . All components with trained utility posterior  $\tilde{\pi} \triangleq q(z=1) < \tau$  are removed from the network; all rest are retained and used at prediction time. In the following tables, the *compression* metric corresponds to the ratio of the number of network components removed to the total number of network components.

We choose an uninformative Beta prior for the IBP (Eq.(14)):  $\text{Beta}(1, 1)$ ; thus,  $\omega = 1$ . For the Gumbel-Softmax relaxation, we set the temperatures  $\tau$  for the prior and posterior distributions to 0.5 and 0.67 respectively (Maddison, Mnih, and Teh 2017). Further initialization procedures and hyperparameter values can be found in the Supplementary.

We draw a single (reparameterized) sample from all the involved random variables during training, while we draw 5 different samples during inference (and proceed with Bayesian averaging). We perform multiple evaluations for each architecture and dataset, i.e., 5 runs, and report the best performing one. All experiments were run on a workstation with 2x Quadro P6000 24GB GPUs and 64GB RAM.

## Experimental Results

Model	VGG-16	GoogLeNet	ResNet20	DenseNet40
	Error (%)		Compression (%)	
Baseline	6.67    0.00	4.92    0.00	7.63    0.00	5.83    0.00
VIB (Alemi et al. 2017)	6.81    0.00	5.09    0.00	6.95    0.00	5.72    0.00
DIM*(Hjelm et al. 2019b)	6.54    0.00	4.65    0.00	7.61    0.00	6.15    0.00
VIB $\times 2$	6.86    0.00	4.88    0.00	6.85    0.00	6.36    0.00
DIM $\ast \times 2$	7.24    0.00	4.95    0.00	7.46    0.00	5.60    0.00
ICP_ALL	6.97    0.00	4.76    0.00	6.47    0.00	6.13    0.00
ICP_COM	6.59    0.00	4.67    0.00	7.33    0.00	5.63    0.00
ICP	6.10    0.00	<b>4.26</b>    0.00	6.01    0.00	4.99    0.00
ICP <sub>IBP</sub> & LWTA-2	<b>6.01</b>    <b>40.4</b>	4.31    35.2	<b>5.94</b>    37.1	<b>4.78</b>    <b>32.0</b>
ICP <sub>IBP</sub> & LWTA-4	7.02    30.4	4.74    28.2	6.30    31.3	5.61    26.3

Table 1: CIFAR-10.

**CIFAR-10.** We train the considered four networks adopting the number of epochs and the optimization parameters used in the original ICP implementation (Hu et al. 2019); this ensures transparency and comparability of the empirical results. The exact experimental setup for each network can be found in the Supplementary. The obtained comparative results for all considered methods and networks can be found in Table 1. Therein, *Baseline* corresponds to the original architectures without any type of MI constraints; VIB $\times 2$  and DIM $\ast \times 2$  denote the VIB and DIM\* methods where the representation dimension has been expanded in order to facilitate a fair (size-wise) comparison with ICP. As we observe, our proposed method yields competitive (in many cases better) classification accuracy over the best-performing baseline ICP alternative. At the same time, by exploiting the IBP-based mechanism, it allows for powerful compression of the considered architecture; this is effected by removing all network components (and associated weights) with trained utility posteriors below the cut-off threshold. Clearly, the aim of this work is not to propose a method for deep network compression. We posit, though, that this compression process further facilitates the diversification of the learned representations. We explore these aspects in Section 4.4.

**CIFAR-100.** For CIFAR-100, we follow an analogous procedure. The classification results are shown in Table 2. In this set of experiments, the increased classification capabilities of the baseline ICP model, that is the conventional ICP model with ReLU-based nonlinearities, compared to the other information-theoretic approaches is more evident. Our proposed approach, once again follows this trend, yielding

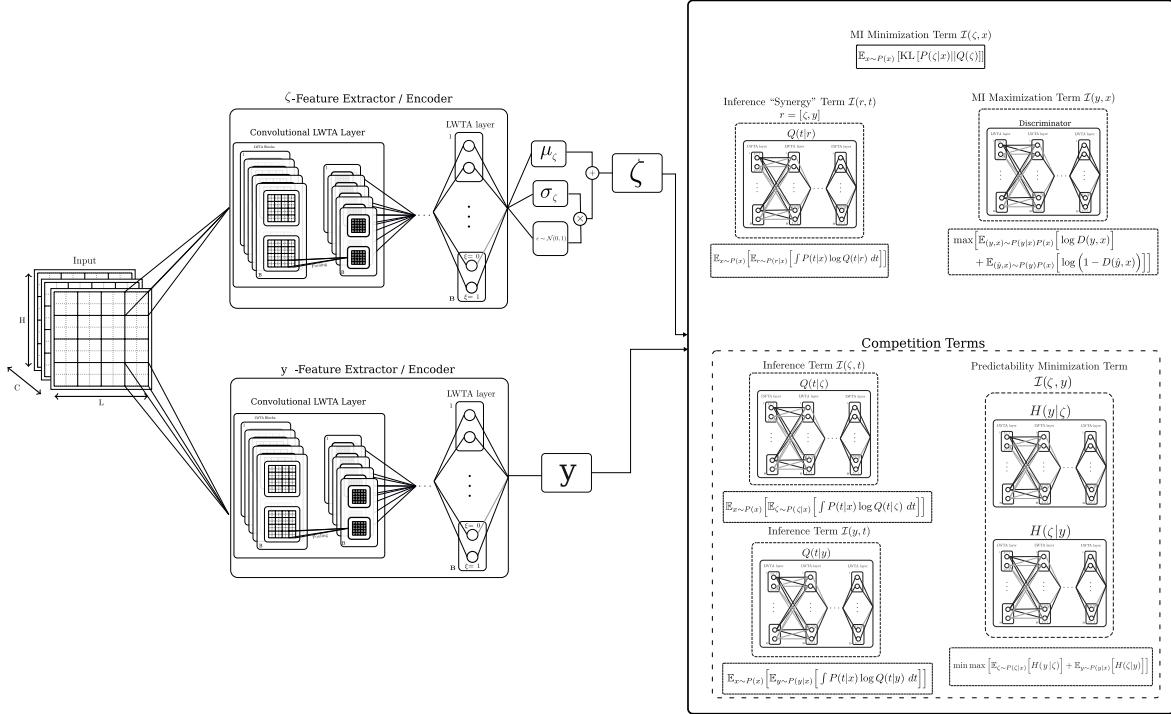


Figure 4: The ICP pipeline: The representation  $r$  is split into two parts  $[\zeta, y]$ . These, not only cooperate but also compete to accomplish the downstream task. This leads to the computation of the MI between different pairs of variables; this is usually intractable and optimized through different schemes. All components of these, are implemented via LWTA and IBP-based DNNs.

on-par or better classification accuracy with ICP, while at the same time exhibiting significant compression capabilities.

Model	VGG-16	GoogLeNet	ResNet20	DenseNet40
	Error (%)    Compression (%)			
Baseline	26.41    0.00	20.68    0.00	31.91    0.00	27.55    0.00
VIB (Alemi et al. 2017)	26.56    0.00	20.93    0.00	30.84    0.00	26.37    0.00
DIM* (Hjelm et al. 2019b)	26.74    0.00	20.94    0.00	32.62    0.00	27.51    0.00
VIB $\times_2$	26.08    0.00	22.09    0.00	29.74    0.00	29.33    0.00
DIM* $\times_2$	25.72    0.00	21.74    0.00	30.16    0.00	27.15    0.00
ICP_ALL	26.73    0.00	20.90    0.00	28.35    0.00	27.51    0.00
ICP_COM	26.37    0.00	20.81    0.00	32.76    0.00	26.85    0.00
ICP	24.54    0.00	<b>18.55</b>    0.00	28.13    0.00	24.52    0.00
ICP <sub>IBP &amp; LWTA-2</sub>	<b>24.35</b>    <b>32.0</b>	19.00    <b>29.0</b>	<b>28.02</b>    <b>31.5</b>	<b>24.44</b>    <b>29.2</b>
ICP <sub>IBP &amp; LWTA-4</sub>	25.44    22.0	20.12    26.5	29.34    21.2	25.07    24.4

Table 2: CIFAR-100.

**Random Seed Effect.** A core aspect of the considered approach lies in its stochastic nature. Thus, in order to assess the overall robustness of the proposed method, we run each experiment 5 times and report the means and standard deviations; these are presented in Table 3. We observe that our approach exhibits consistent performance in all cases, while in most occasions, the mean performance obtained by multiple runs outperforms the baseline ICP approach.

### Ablation Study

Here, we focus on the VGG-16 architecture described in the previous section, and assess the individual impact of each of the proposed components, i.e. LWTA and IBP, to the classification performance of the network. At the same time, we

CIFAR-10				
Model	Mean (%)    Standard Deviation (%)		ResNet20	DenseNet40
	VGG-16	GoogLeNet		
ICP <sub>IBP &amp; LWTA-2</sub>	6.07    0.04	4.35    0.04	6.00    0.06	4.87    0.05
ICP <sub>IBP &amp; LWTA-4</sub>	7.12    0.08	4.84    0.06	6.42    0.10	5.70    0.08

CIFAR-100				
Model	Mean (%)    Standard Deviation (%)		ResNet20	DenseNet40
	VGG-16	GoogLeNet		
ICP <sub>IBP &amp; LWTA-2</sub>	24.52    0.12	19.25    0.14	28.10    0.05	24.52    0.05
ICP <sub>IBP &amp; LWTA-4</sub>	25.59    0.08	20.28    0.10	29.51    0.10	25.19    0.08

Table 3: Means and standard deviations of 5 different runs under different seeds for all datasets and architectures.

examine whether adoption of a deterministic LWTA scheme, as opposed to the adopted stochastic construction, would yield equal or inferior performance in these benchmarks.

The obtained comparative results are depicted in Table 4. Therein, the  $ICP_{LWTA-*max}$  and  $ICP_{IBP \& LWTA-*max}$  entries correspond to networks where the LWTA competition function picks the unit with greatest activation value, in a deterministic fashion;  $* = 2, 4$  denotes the number of competitors in each LWTA block. We consider both omission and use of the IBP-based mechanism, respectively.

As we observe, for both  $U = 2, 4$  settings, the Stochastic LWTA approach outperforms deterministic LWTA units picking the greatest value. The IBP-based mechanism seems to also facilitate classification accuracy (in addition to compressing the network we perform inference with).

Table 4: Ablation Study: CIFAR-10 test-set using a VGG-16 (Simonyan and Zisserman 2015) architecture.

Model	VGG-16
Baseline	6.67
ICP	6.10
$\text{ICP}_{\text{LWTA-2}^{\max}}$	6.34
$\text{ICP}_{\text{LWTA-2}}$	6.23
$\text{ICP}_{\text{IBP \& LWTA-2}^{\max}}$	6.27
$\text{ICP}_{\text{IBP \& LWTA-2}}$	<b>6.01</b>
$\text{ICP}_{\text{LWTA-4}^{\max}}$	7.01
$\text{ICP}_{\text{LWTA-4}}$	6.85
$\text{ICP}_{\text{IBP \& LWTA-4}^{\max}}$	7.32
$\text{ICP}_{\text{IBP \& LWTA-4}}$	7.02

## Representation Diversification

The ultimate goal of this work is to allow for deep networks to yield representations that are sufficiently diverse. In this Section, we investigate the diversification capabilities of the proposed framework, both qualitatively and quantitatively. To perform this analysis, and due to space limitations, we focus on the VGG-16 architecture and the CIFAR-10 dataset. Similar results on further architectures and datasets can be found in the Appendix.

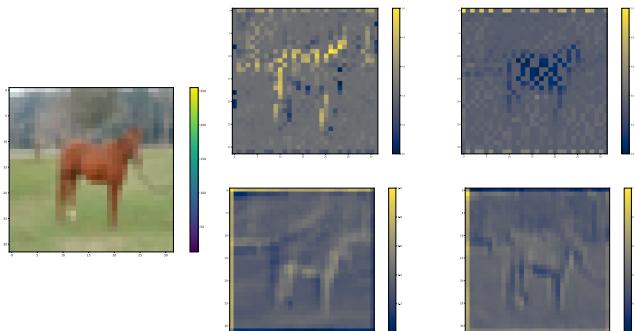


Figure 5: Feature map visualizations for a test example of the CIFAR-10 dataset, emerging from the first layer of the considered VGG-16 architecture. (Left) The original image, (Upper Row) a visualization of the outputs of the two competing feature maps of the first LWTA block ( $U = 2$ ), and (Bottom Row) a visualization of the first two filters of the conventional ReLU-based approach. The latter exhibit significant overlap, contrary to the much diverse representations of our approach which are mutually-exclusive.

To perform a qualitative evaluation, we visually compare the emerging *intermediate representations from the hidden layers* of employed ReLU-based networks (standard ICP) and our approach. As we observe in Fig. 5, there exists a clear disparity between the proposed approach and the commonly employed nonlinearities. Clearly, the ReLU-based architecture allows for more “aesthetically pleasing” representations. However, the proposed networks yield significant *diversification*, as they split the resulting representations from each

block to *mutually exclusive parts*. This constitutes a radically different RL scheme with significant diversification capacity for the emerging *intermediate* representations.

For the quantification of the diversification properties, we turn to the commonly used Linear Separability metric. This metric constitutes a proxy for quantifying the disentanglement and MI between the emerging representations and the class labels (Hjelm et al. 2019a). In this context, linear classification is usually considered, and for this we use the standard Support Vector Machine (SVM) approach. To this end, we hold the encoders of  $\mathbf{y}$  and  $\zeta$  fixed, and build separate SVM-based classifiers on the two representation parts  $\zeta$ ,  $\mathbf{y}$ , their combination (denoted as Total), as well as the output of the last convolutional layer of the  $\zeta$  encoder (Conv), which stems either from ReLU units or LWTA blocks.

The obtained comparative separability results are depicted in Table 5. We observe that, by training a linear SVM model on the obtained representations  $\zeta$  or  $\mathbf{y}$ , their combination thereof, or the last convolutional layer of the underlying encoder, the linear model can obtain more potent accuracy profile if its inputs stem from networks formulated under our approach (LWTA & IBP arguments). This hints at obtaining more diversified representations, that therefore convey richer information to the linear SVM classifier, which is also easier to discern in a linear classification fashion.

Table 5: Results on linear separability using SVMs.

Method	Proxies			
	SVM( $\zeta$ )	SVM( $\mathbf{y}$ )	SVM(Total)	SVM(Conv)
ICP	91.5	91.9	92.9	31.2
$\text{ICP}_{\text{LWTA-2 \& IBP}}$	<b>91.9</b>	<b>92.4</b>	<b>93.4</b>	<b>32.2</b>

## 5 Conclusions

In this work, we attacked the problem of promoting diversified representations in Deep Learning. To this end, we introduced principled network arguments formulated by stochastic competition-based Local Winner-Takes-All activations. We combined these with network component utility mechanisms, which rely on the use of the IBP prior. Then, to further enrich the emerging representations, we employed information-theoretic arguments, founded on competing mutual information constraints under the Information Competing Process. This results in an efficient network training and prediction scheme, that significantly compresses the networks during prediction. We performed a thorough experimental evaluation, using benchmark datasets and several standard network architectures. We compared networks crafted using the proposed arguments against standard, ReLU-based constructions. Our experimental results provided strong empirical evidence of the efficacy of the proposed framework. Specifically, in all cases, our approach yielded on-par or improved accuracy for significantly compressed networks. At the same time, our qualitative and quantitative analysis of the obtained representations showed our approach results in representations that: (i) visually appear much more diverse; and (ii) are more informative to a linear classifier trained on them, specifically an SVM used as a proxy to linear separability.

## Acknowledgements

This work has received funding from the European Union’s Horizon 2020 research and innovation program under grant agreement No 872139, project aiD.

## References

- Achille, A.; and Soatto, S. 2017. Information Dropout: Learning Optimal Representations Through Noisy Computation. arXiv:1611.01353.
- Alemi, A.; Fischer, I.; Dillon, J.; and Murphy, K. 2017. Deep Variational Information Bottleneck. In *ICLR*.
- Alemi, A.; Poole, B.; Fischer, I.; Dillon, J.; Saurous, R. A.; and Murphy, K. 2018. Fixing a Broken ELBO. In *Proc. ICML*.
- Alemi, A. A.; Fischer, I.; and Dillon, J. V. 2018. Uncertainty in the Variational Information Bottleneck. arXiv:1807.00906.
- Andersen, P.; Gross, G. N.; Lomo, T.; and Sveen, O. 1969. Participation of inhibitory and excitatory interneurones in the control of hippocampal cortical output. In *UCLA forum in medical sciences*.
- Belghazi, M. I.; Baratin, A.; Rajeswar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, R. D. 2018. MINE: Mutual Information Neural Estimation. arXiv:1801.04062.
- Bell, A.; and Sejnowski, T. 1995. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. *Neural computation*.
- Carpenter, G. A.; and Grossberg, S. 1988. The ART of adaptive pattern recognition by a self-organizing neural network. *Computer*.
- Dai, B.; Zhu, C.; and Wipf, D. 2018. Compressing Neural Networks using the Variational Information Bottleneck. arXiv:1802.10399.
- Douglas, R. J.; and Martin, K. A. 2004. Neuronal circuits of the neocortex. *Annu. Rev. Neurosci.*
- Ghahramani, Z.; and Griffiths, T. L. 2006. Infinite latent feature models and the Indian buffet process. In *Proc. NIPS*.
- Greff, K.; Rasmus, A.; Berglund, M.; Hao, T.; Valpola, H.; and Schmidhuber, J. 2016. Tagger: Deep Unsupervised Perceptual Grouping. In *Proc. NIPS*.
- Greff, K.; van Steenkiste, S.; and Schmidhuber, J. 2017. Neural Expectation Maximization. In *Proc. NIPS*.
- Grossberg, S. 1982. Contour enhancement, short term memory, and constancies in reverberating neural networks. In *Studies of mind and brain*. Springer.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proc. CVPR*.
- Higgins, I.; Matthey, L.; Pal, A.; Burgess, P. C.; Glorot, X.; Botvinick, M.; Mohamed, S.; and Lerchner, A. 2017. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework. *Proc. ICLR*.
- Hjelm, D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2019a. Learning deep representations by mutual information estimation and maximization. In *Proc. ICLR*.
- Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2019b. Learning deep representations by mutual information estimation and maximization. In *Proc. ICLR*.
- Hu, J.; Ji, R.; Zhang, S.; Sun, X.; Ye, Q.; Lin, C.-W.; and Tian, Q. 2019. Information Competing Process for Learning Diversified Representations. In *Proc. NIPS*.
- Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely Connected Convolutional Networks. In *Proc. CVPR*.
- Jang, E.; Gu, S.; and Poole, B. 2017. Categorical Reparametrization with Gumbel-Softmax. In *Proc ICLR*.
- Kandel, E. R.; Schwartz, J. H.; and Jessell, T. M. 2000. *Principles of Neural Science*. Elsevier.
- Kingma, D. P.; and Welling, M. 2014. Auto-Encoding Variational Bayes. In *Proc. ICLR*.
- Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto.
- Kumaraswamy, P. 1980. A generalized probability density function for double-bounded random processes. *Journal of Hydrology*.
- Lansner, A. 2009. Associative memory models: from the cell-assembly theory to biophysically detailed cortex simulations. *Trends in neurosciences*.
- Linsker, R. 1988. Self-organization in a perceptual network. *Computer*.
- Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *Proc ICLR*.
- Panousis, K.; Chatzis, S.; and Theodoridis, S. 2019. Non-parametric Bayesian Deep Networks with Local Competition. In *Proc. ICML*.
- Panousis, K.; Chatzis, S.; and Theodoridis, S. 2021. Stochastic Local Winner-Takes-All Networks Enable Profound Adversarial Robustness. In *Bayesian Deep Learning NeurIPS workshop*.
- Panousis, K. P.; Chatzis, S.; Alexos, A.; and Theodoridis, S. 2021. Local Competition and Stochasticity for Adversarial Robustness in Deep Learning. In *Proc. AISTATS*.
- Ranjan, A.; Jampani, V.; Balles, L.; Kim, K.; Sun, D.; Wulff, J.; and Black, M. J. 2019. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation. In *Proc. CVPR*.
- Saxe, A. M.; Bansal, Y.; Dapello, J.; Advani, M.; Kolchinsky, A.; Tracey, B. D.; and Cox, D. D. 2018. On the Information Bottleneck Theory of Deep Learning. In *Proc. ICLR*.
- Shwartz-Ziv, R.; and Tishby, N. 2017. Opening the Black Box of Deep Neural Networks via Information. *ArXiv*.
- Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proc. ICLR*.
- Srivastava, R. K.; Masci, J.; Kazerounian, S.; Gomez, F.; and Schmidhuber, J. 2013. Compete to compute. In *Proc. NIPS*.
- Stefanis, C. 1969. Interneuronal mechanisms in the cortex. In *UCLA forum in medical sciences*.

- Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proc. CVPR*.
- Teh, Y. W.; Grür, D.; and Ghahramani, Z. 2007. Stick-breaking construction for the Indian buffet process. In *Proc. AISTATS*.
- Tishby, N.; Pereira, F. C.; and Bialek, W. 1999. The information bottleneck method. In *Proc. of the 37-th Annual Allerton Conference on Communication, Control and Computing*.
- Tishby, N.; and Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*.
- van den Oord, A.; Li, Y.; and Vinyals, O. 2019. Representation Learning with Contrastive Predictive Coding. arXiv:1807.03748.
- Voskou, A.; Panousis, K. P.; Kosmopoulos, D.; Metaxas, D. N.; and Chatzis, S. 2021. Stochastic Transformer Networks With Linear Competing Units: Application To End-to-End SL Translation. In *Proc. ICCV*.