Supplemental Material
DISCOVER:
Making Vision Networks Interpretable via Competition and Dissection

Konstantinos Panousis
Department of Electrical Eng., Computer Eng., and Informatics
Cyprus University of Technology
Limassol 3036, Cyprus
k.panousis@cut.ac.cy

Sotirios Chatzis
Department of Electrical Eng., Computer Eng., and Informatics
Cyprus University of Technology
Limassol 3036, Cyprus
sotirios.chatzis@cut.ac.cy

A Experimental Details

Integrating the competition mechanism in modern architectures constitutes a pretty straightforward task. One just needs to replace the existing calls to the conventional non-linear activations with a call to a module/function implementing the LWTA rationale. This can be performed in-place without any other changes necessary. Thus, in the accompanied code, we can directly change the definitions of the Transformer or ResNet based models and train the model in the same manner as in the conventional Vision Networks. In this context, for the DeiT architectures, we alter the definitions in the *timm* library, while for ResNet-based architectures, we slightly alter the example implementation found in the official Pytorch repository[1]. For dissecting CVNs, we use the official repository of CLIP-Dissect [3]. We describe all changes in the respective README file. Our code and trained models will be public after publication.

Transformers. For training the CVN counterparts of the DeiT-T and DeiT-S models, we use the official implementation.[2] We train both architectures from scratch using ImageNet-1k for 300 epochs with the default parameters found therein. Specifically, we use a 5-epoch warm-up period, starting with an initial learning rate of 10^{-6}, following a cosine annealing schedule up to $5 \cdot 10^{-4}$. We use the same AdamW optimizer and changed the used weight decay from 0.05 to 0.02 since we found that it hurt performance. We re-run the conventional GELU-based architectures with this selection and observed no change in the obtained accuracy. In contrast, we turned off the excessive augmentation setup, since it hurt the performance of CVNs. In this context, and in line with the initial ablation study presented in [7], we found that performance deteriorates when augmentations are removed in GELU-based networks. For training, we use a single sample for the Monte Carlo sampling estimation for the Evidence Lower Bound loss, described in the main text. For this, we turn to the continuous relaxation of the categorical distribution that allows for reparameterized samples and low variance estimation, as we describe in the next section. During inference, we draw 4 samples,

---

[1] `https://github.com/pytorch/examples/tree/main/imagenet`

[2] `https://github.com/facebookresearch/deit`.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

average the logits in the Bayesian Averaging sense; we then compute the predicted loss and accuracy. We did not observe any improvement when drawing more samples.

ResNet. For training the ResNet-18 model, we used the ResNet ImageNet example PyTorch script and adapted the data loading for Places 365. We train the model for 90 epochs, using SGD with an initial learning rate of 0.1 that is reduced by a factor of 0.1 every 30 epochs, a weight decay of 10^{-4} and 0.9 momentum. The batch size was set to 256. For the Gumbel-Softmax trick, we set the temperature to 0.67 and used the Straight-Through estimator. During training, we only draw one sample for the reparameterization trick, while during inference we draw 4 samples from the trained posterior. We trained both conventional and CVN architectures since the official pretrained models were not available online. For both methods, we used the standard RandomResizedCrop and RandomHorizontal Flip augmentations.

A.1 The Gumbel-Softmax Trick

In our work, we perform Monte-Carlo sampling using a single reparameterized sample for each of the corresponding latent variables. These are obtained via the reparameterization trick of the continuous relaxation of the Categorical distribution [2, 1] as described next. We focus on the reparameterization trick for the dense case; the convolutional case is analogous.

As a reminder, the latent indicators $\boldsymbol{\xi}_b, \forall b$, are drawn from a Categorical distribution driven from the intermediate linear inner-product computations that each unit performs.of $q(\boldsymbol{\xi})$ (Eq. (2) in the main text):

$$q(\boldsymbol{\xi}_b) = \text{Categorical}\left(\boldsymbol{\xi}_b \middle| \Pi_b(\boldsymbol{x})\right), \forall b, \quad \boldsymbol{\Pi}_b(\boldsymbol{x}) = \text{Softmax}\left(\sum_{j=1}^{J} [w_{j,b,u}]_{u=1}^{U} \cdot x_j\right) \tag{1}$$

where $\Pi_b(\boldsymbol{x})$ is a vector comprising the *activation probability* of each neuron in the block, and $[w_{j,b,u}]_{u=1}^{U}$ denotes the vector concatenation of the set $\{w_{j,b,u}\}_{u=1}^{U}$.

Then, the samples $\hat{\boldsymbol{\xi}}$ are expressed as:

$$\hat{\xi}_{b,u} = \text{Softmax}\left(\log[\boldsymbol{\Pi}_b(\boldsymbol{x})]_u + g_{b,u})/\tau\right), \forall b = 1, \dots, B, \ u = 1, \dots, U \tag{2}$$

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 factor, controlling how "closely" the continuous relaxation approximates the Categorical distribution. In this work, we use a temperature of $\tau = 0.67$ as suggested in [2], and used in several other works [5, 4].

A.2 Convolutional Formulation

In this setting, local competition is performed among feature maps on a *position-wise* basis. Each kernel is treated as an LWTA block with competing feature maps; each layer comprises B kernels. Specifically, each feature map $u = 1, \dots, U$ in the b^{th} LWTA block of a convolutional LWTA layer computes:

$$\boldsymbol{H}_{b,u} = \boldsymbol{W}_{b,u} \star \boldsymbol{X} \in \mathbb{R}^{H \times L} \tag{3}$$

Competition remains stochastic, and is now implemented on a *position-wise* basis as follows:

$$q(\boldsymbol{\xi}_{b,h',l'}) = \text{Categorical}\left(\boldsymbol{\xi}_{b,h',l'} \middle| \boldsymbol{\Pi}_{b,h',l'}(\boldsymbol{X})\right), \forall h', l' \tag{4}$$

where $\boldsymbol{\Pi}_{b,h',l'}(\boldsymbol{X}) = \text{Softmax}\left([\boldsymbol{H}_{b,1,h',l'}, \dots, \boldsymbol{H}_{b,U,h',l'}]\right)$ comprises the *position-wise activation probabilities* for all neurons in the block.

For each position in a kernel, only the feature map that wins the said position contains a non-zero entry. This yields sparse feature maps with mutually exclusive activated positions. Now, the output $\boldsymbol{Y} \in \mathbb{R}^{H \times L \times B \cdot U}$ is obtained via concatenation of the sub-tensors $\boldsymbol{Y}_{b,u}$ that read:

$$\boldsymbol{Y}_{b,u} = \boldsymbol{\Xi}_{b,u}\Big(\boldsymbol{W}_{b,u} \star \boldsymbol{X}\Big), \ \forall b, u \tag{5}$$

where $\Xi_{b,u} = [\xi_{b,u,h',l'}]_{h',l'=1}^{H,L}$. The corresponding detailed bisection of a convolutional stochastic LWTA block can be found in the Appendix.

In the following, we use these definitions to construct Competitive Vision Networks by replacing the usually employed non-linearities with the competition mechanism in each hidden layer of the considered architecture.

B Further Qualitative Analysis

In this section, we provide further qualitative neuron identification results for various architectures and similarity functions.

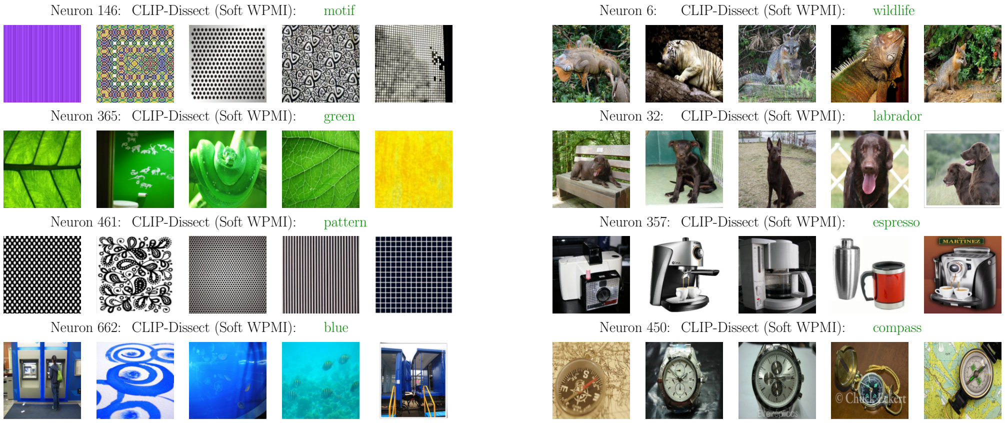
Neuron 146: CLIP-Dissect (Soft WPMI): motif
Neuron 365: CLIP-Dissect (Soft WPMI): green
Neuron 461: CLIP-Dissect (Soft WPMI): pattern
Neuron 662: CLIP-Dissect (Soft WPMI): blue
Neuron 6: CLIP-Dissect (Soft WPMI): wildlife
Neuron 32: CLIP-Dissect (Soft WPMI): labrador
Neuron 357: CLIP-Dissect (Soft WPMI): espresso
Neuron 450: CLIP-Dissect (Soft WPMI): compass

Figure 2: Neuron Identification for the first and last DeiT-T/8 MLP blocks using: SoftWPMI, $\mathcal{D}_{\text{probe}}$: ImageNet & Broden, $\mathcal{S}$: $20K$ most common English words[3].

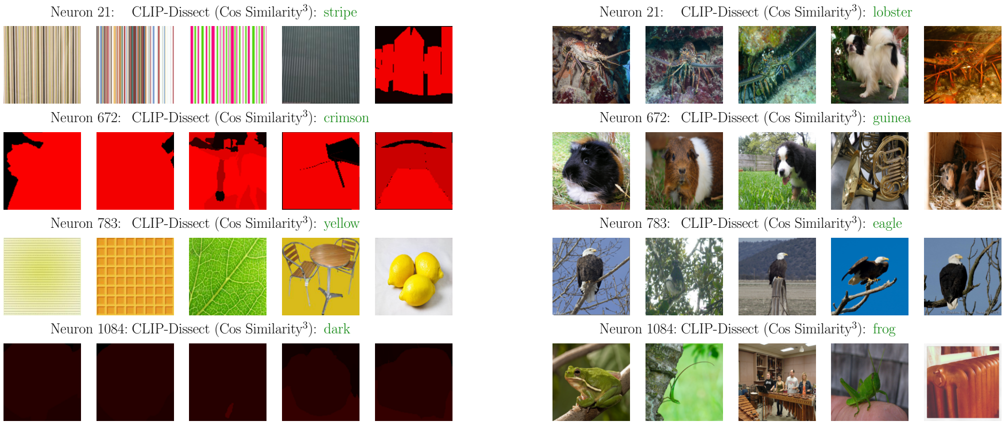
Neuron 21: CLIP-Dissect (Cos Similarity^3): stripe
Neuron 672: CLIP-Dissect (Cos Similarity^3): crimson
Neuron 783: CLIP-Dissect (Cos Similarity^3): yellow
Neuron 1084: CLIP-Dissect (Cos Similarity^3): dark
Neuron 21: CLIP-Dissect (Cos Similarity^3): lobster
Neuron 672: CLIP-Dissect (Cos Similarity^3): guinea
Neuron 783: CLIP-Dissect (Cos Similarity^3): eagle
Neuron 1084: CLIP-Dissect (Cos Similarity^3): frog

Figure 3: Neuron Identification for the first and last DeiT-S/12 MLP block using the Cosine Similarity Cubed similarity function [3]. For the former we use $\mathcal{D}_{\text{probe}}$: (i) ImageNet Val & Broden (Left), and for the latter: (ii) ImageNet Val. We use the $\mathcal{S}$: $20K$ most common English words for both settings.

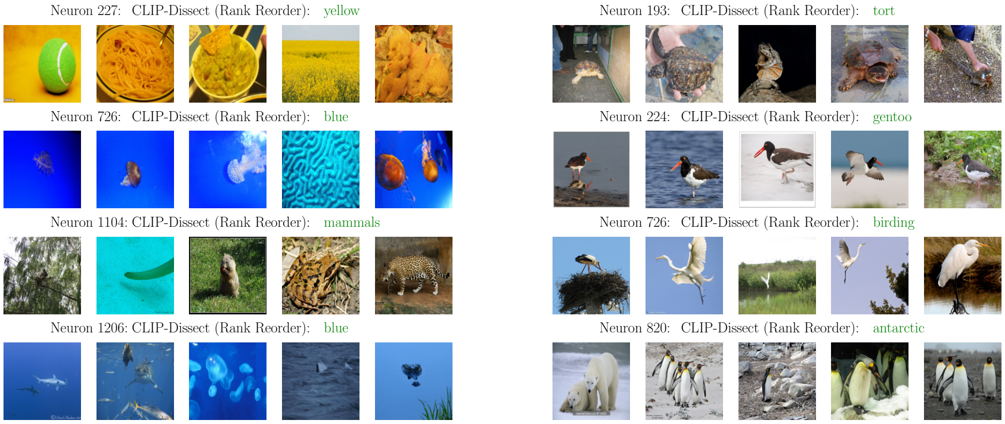
Neuron 227: CLIP-Dissect (Rank Reorder): yellow
Neuron 726: CLIP-Dissect (Rank Reorder): blue
Neuron 1104: CLIP-Dissect (Rank Reorder): mammals
Neuron 1206: CLIP-Dissect (Rank Reorder): blue
Neuron 193: CLIP-Dissect (Rank Reorder): tort
Neuron 224: CLIP-Dissect (Rank Reorder): gentoo
Neuron 726: CLIP-Dissect (Rank Reorder): birding
Neuron 820: CLIP-Dissect (Rank Reorder): antarctic

Figure 4: Neuron Identification for the first and last DeiT-S/12 MLP block using the Rank Reorder similarity function [3]. We use $\mathcal{D}_{\text{probe}}$: ImageNet Val and $\mathcal{S}$: $20K$ most common English words for both settings.

Table 1: Concepts tied to activated neurons for a random image from the ImageNet validation set for DeiT-T/8 using various similarity functions.

Cos Similarity		WPMI		SoftWPMI	
Concepts	Act	Concepts	Act	Act	Concepts
Belgian Malinois	$+10.5$	ferret	$+10.5$	ferret	$+10.5$
green/yellow eyes	$+9.55$	ferret	$+9.55$	ferret	$+9.55$
white fur underside	$+8.90$	white fur under tail	$+8.90$	white fur under tail	$+8.90$
herd Alpine ibex	$+8.51$	black stripes on legs	$+8.51$	black/white stripes	$+8.51$
long/sharp quills	$+8.02$	long/sharp quills	$+8.01$	long/sharp quills	$+8.01$
gray fur/white tips	$+7.53$	fox	$+7.53$	fox	$+7.53$
white/gray head	$+7.52$	white/gray head	$+7.52$	white/gray head	$+7.52$
baby stork	$+0.96$	black ruff around neck	$+1.09$	black ruff around neck	$+1.09$
black fur/white markings	$+0.44$	black/tan coloration	$+0.44$	black/tan coloration	$+0.44$
large/fluffy white dog	$+0.35$	organ	$+0.37$	organ	$+0.37$
gills	$+0.18$	salamander	$+0.18$	salamander	$+0.18$
car towing RV	$+0.12$	headsail	$+0.12$	headsail	$+0.12$
other crocodiles	$+0.12$	egg-laying mammal	$+0.12$	egg-laying mammal	$+0.12$

Table 2: Concepts tied to activated neurons for a random image from the ImageNet validation set for DeiT-T/None and DeiT-S/12 using various similarity functions.

DeiT-T/None (SoftWPMI)		DeiT-S/12 (SoftWPMI)		DeiT-S/12 (Rank Reorder)	
Concepts	Act	Concepts	Act	Act	Concepts
a coast	$+1.18$	sea turtle	$+15.0$	sea turtle	$+15.2$
hard/shiny exoskeleton	$+1.06$	cygnet	$+11.7$	large/webbed hind feet	$+11.7$
sea turtle	$+0.54$	large/elephant animal	$+10.9$	dark-colored carapace	$+10.9$
large/red crab	$+0.54$	hard shell covered in spines	$+9.96$	large/fleshy cap	$+9.96$
large/red crustacean	$+0.53$	sea turtle	$+9.52$	marine ecosystem	$+9.52$
large/plant eating dinosaur	$+0.51$	python	$+9.32$	tough/scaly exterior	$+9.32$
short/bristly fur	$+0.46$	doberman	$+9.08$	black/tan coloration	$+9.08$
white bill	$+0.20$	dome-shaped cap	$+1.14$	round/spouted body	$+1.41$
a race	$+0.17$	iguana	$+0.63$	fluffy/white appearance	$+0.33$
large/fluffy white dog	$+0.06$	other crocodiles	$+0.67$	operating system	$+0.22$
two-metal plates/trays	-0.16	case	$+0.22$	opponent	$+0.10$
white markings wings	-0.16	players	$+0.10$	wings attached	$+0.10$
small/crab like body	-0.17	exoskeleton	$+0.01$	hard/shiny exoskeleton	$+0.10$

References

[1] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumbel-softmax. In *Proc. ICLR*, 2017.

[2] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proc. ICLR*, 2017.

[3] Tuomas Oikarinen and Tsui-Wei Weng. CLIP-dissect: Automatic description of neuron representations in deep vision networks. In *Proc. ICLR*, 2023.

[4] Konstantinos Panousis, Sotirios Chatzis, Antonios Alexos, and Sergios Theodoridis. Local competition and stochasticity for adversarial robustness in deep learning. In *Proc. AISTATS*, 2021.

[5] Konstantinos Panousis, Sotirios Chatzis, and Sergios Theodoridis. Nonparametric Bayesian deep networks with local competition. In *Proc. ICML*, 2019.

[6] Konstantinos P Panousis, Anastasios Antoniadis, and Sotirios Chatzis. Competing mutual information constraints with stochastic competition-based activations for learning diversified representations. In *Proc. AAAI*, 2022.

[7] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In *Proc. ICML*, 2021.